agriculture

Article

Fused Deep Features-Based Grape Varieties Identification Using Support Vector Machine

Yun Peng [1,2], Shenyi Zhao [1] and Jizhan Liu [1,*]

[1] Key Laboratory of Modern Agricultural Equipment and Technology, Ministry of Education, Jiangsu University, Zhenjiang 212013, China; 2111716004@stmail.ujs.edu.cn (Y.P.); 2111916017@stmail.ujs.edu.cn (S.Z.)
[2] School of Electronic Engineering, Changzhou College of Information Technology, Changzhou 213164, China
[*] Correspondence: 1000002048@ujs.edu.cn; Tel.: +86-511-88797338

Abstract: Proper identification of different grape varieties by smart machinery is of great importance to modern agriculture production. In this paper, a fast and accurate identification method based on Canonical Correlation Analysis (CCA), which can fuse different deep features extracted from Convolutional Neural Network (CNN), plus Support Vector Machine (SVM) is proposed. In this research, based on an open dataset, three types of state-of-the-art CNNs, seven species of deep features, and a multi-class SVM classifier were studied. First, the images were resized to meet the input requirements of a CNN. Then, the deep features of the input images were extracted by a specific deep features layer of the CNN. Next, two kinds of deep features from different networks were fused by CCA to increase the effective classification feature information. Finally, a multi-class SVM classifier was trained with the fused features. When applied to an open dataset, the model outcome shows that the fused deep features with any combination can obtain better identification performance than by using a single type of deep feature. The fusion of fc6 (in AlexNet network) and Fc1000 (in ResNet50 network) deep features obtained the best identification performance. The average F1 Score of 96.9% was 8.7% higher compared to the best performance of a single deep feature, i.e., Fc1000 of ResNet101, which was 88.2%. Furthermore, the F1 Score of the proposed method is 2.7% higher than the best performance obtained by using a CNN directly. The experimental results show that the method proposed in this paper can achieve fast and accurate identification of grape varieties. Based on the proposed algorithm, the smart machinery in agriculture can take more targeted measures based on the different characteristics of different grape varieties for further improvement of the yield and quality of grape production.

Keywords: grape varieties identification; Support Vector Machine (SVM); Convolutional Neural Network (CNN); deep feature fusion; Canonical Correlation Analysis (CCA); smart machinery

check for updates

Citation: Peng, Y.; Zhao, S.; Liu, J. Fused Deep Features-Based Grape Varieties Identification Using Support Vector Machine. *Agriculture* **2021**, *11*, 869. https://doi.org/10.3390/agriculture11090869

Academic Editor: Dimitre Dimitrov

Received: 10 August 2021
Accepted: 8 September 2021
Published: 10 September 2021

1. Introduction

Grape is one of the most popular fruits which can be used for wine production or fresh food. There are many varieties of grapes in the world, with more than 10,000 varieties and about 3000 cultivars. Through the improvement and screening of grape varieties, there are dozens of wine grapes widely planted at present. Many wine producers need to mix a larger number of different varieties of grapes to produce high-quality wines, such as the "Blend D. Antónia", which has more than 30 grape varieties. This means that more than one grape variety is planted per parcel and even per row [1]. Although the basic management methods of grapes are similar, the different varieties have their own characteristics, and they have different requirements for pruning, spraying, fertilization, and harvest time. Scientific and accurate field management is the key to the production of wine grapes of high quality. With the development of modern agriculture, more smart machines are used for pruning [2], spraying [3,4], and harvesting grapes [5]. Accurate identification of grape varieties is necessary for the smart machinery to make more targeted decisions for different varieties. Therefore, it is important to develop reliable methods that could automatically identify the varieties of grapes.

The traditional methods of grape variety identification mainly include manual and chemical identification [6]. Manual identification requires a high level of experience and expertise of the inspector and has disadvantages such as being time-consuming and subjective. In contrast, chemical identification methods such as mass spectrometry, chromatography, and fluorescence spectrometry are widely used, have demonstrated great efficiency, and are considered to be reliable for identifying grape varieties [7–10]. However, it is not convenient to apply the chemical methods to smart machinery, and these methods cannot produce real-time identification results for the real-time operation of smart machinery. Machine vision, which only needs an image sensor, is very easy and convenient to integrate into smart machinery. Thus, in this paper, the method of machine vision was considered to identify grape varieties. With the development of machine vision, image inspection as a non-destructive technology is widely used in industry [11,12], agriculture [13,14], and fisheries [15,16], showing great application prospects.

In the agriculture field, machine vision-based methods are widely used for the classification or grading of agricultural products. Nasirahmadi et al. used a Bag of Features (BoF) model to solve the classification problem of 20 kinds of almonds. In their research, three classifiers (L-SVM, Chi-SVM, and kNN) based on five keypoint detectors and a SIFT descriptor were investigated and achieved 79–87%, 83–91%, and 67–78% accuracy, respectively [17]. Bhargava et al., aiming at the problem of apple (fresh, rotten), six different varieties of apple (Fuji, York, Golden Crown, Red Crown, Granny Smith, and Jonagold) were selected. Firstly, the fruit region in the image was segmented by grab-cut method and Fuzzy C-Means Clustering, and then six features were extracted from feature space by principal component analysis to train an SVM classifier, and finally, 98.42% classification accuracy was obtained [18]. In [19], Ponce et al. focused on the variety identification of olive fruits. Six different Convolutional Neural Networks were trained by 2800 images with seven kinds of olive, and a top accuracy of 95.91% was obtained by Inception-ResnetV2. In addition, there are also some studies about the identification of grape varieties based on machine learning or deep learning. Bogdan et al. developed a model which is a combination of deep learning ResNet50 classifier model with multi-layer perceptron for grape varieties identification. A well-known benchmark dataset, which provided the instances from five different grape varieties taken from the field, was used for training and testing on the developed model. The test results showed that the classification accuracy of the model for different grape varieties can reach 99% [20]. El-Mashharawi et al. adopted a CNN for the identification of grape varieties. A dataset provided by Kaggle, which contains 4565 images with six species of grapes, was used to train and test the network, and a validation accuracy of 100% was achieved on the test set. The main reason for such high accuracy could be attributed to that each image in the dataset contains only one grape grain, and the background is pure white without any interference [21]. Besides, the AlexNet was trained by Pereira et al. with 10 different generated datasets and the highest accuracy of 77.3% was obtained with the four-corners-in-one preprocessed dataset [1].

Two kinds of methods were mainly adopted in the above studies: (1) SVM-based models for classification and (2) deep learning-based models for classification. SVM-based classifiers have a simple classification idea, i.e., to maximize the interval between samples and decision surfaces. Thus, using kernel functions to map features to high-dimensional space to solve nonlinear classification problems can often achieve excellent results [22,23]. However, the classification effect often depends on the class, number, and robustness of the selected features, which is often specific to the research and limited by their experience, and thus, the performance of classifiers trained by different people often has large performance differences. Compared to traditional machine learning methods, deep learning often could achieve better performance by the application of a Convolutional Neural Network (CNN), which consists of a variety of filters, nonlinearities, and pooling operators. The filters with different sizes are used for learning. A nonlinear operator such as hyperbolic tangents, rectified linear units, or logistics sigmoid are added to improve the nonlinear fitting ability of the model. Convolution and nonlinearities are usually followed by a pooling operator such as subsampling, average pooling, or maximum pooling. The CNN model can automatically discover features, and as the number of CNN layers increases, the feature discovered becomes more advanced. Compared to handcraft features by manual selection, deep learning can automatically learn the hierarchical features hidden into the images, which is not only more effective but also avoids the tedious features selection procedure.

However, when dealing with a small dataset, it is difficult to train a CNN from scratch. Based on the discussion in the previous section, it is natural to carry out a method that can train the SVM model with deep features to make full use of the advantages of the two techniques. The literature reports that this method has been applied in agriculture and achieved good performance. In [24], Sethy et al. evaluated the classification performance of a Support Vector Machine classifier for rice disease identification. The features adopted for SVM classifier training were extracted by 13 pre-trained CNN models (AlexNet, Vgg16, Vgg19, Xception, Resnet18, Resnet50, Resnet101, Inceptionv3, Inceptionresnetv2, GoogLeNet, Densenet201, Mobilenetv2, shufflenet). The use of pre-trained models can not only ensure that the model could extract effective features, but at the same time, it avoids the need for large datasets and computational resources to train a CNN. The results show that the classifier trained by features extract by ResNet50 is superior to other models, and the F1 Score is 0.9838. In the same year, this method was used by the team to detect nitrogen deficiency of rice, and the classifier of ResNet50 + SVM achieved the best classification accuracy of 99.84% [25]. In addition, Jiang et al., adopting the same idea as [24,25], developed an SVM classifier for rice leaf diseases diagnosis, and an average correct recognition rate of 96.8% was obtained. The above studies show that training an SVM classifier with the deep features extracted by a CNN model can achieve classification results that are not inferior to those of applying the corresponding deep model directly. Moreover, the computational resources, as well as the training time, are significantly reduced compared with training a deep network from scratch. However, the currently proposed methods of deep features plus SVM have two problems. (1) The dimensionality of the deep features is usually very large; for example, the dimensionality of fc6, fc7, and fc8 of AlexNet are 4096, 4096, and 1000, respectively. The high-dimensional features may affect the classification performance since the SVM model is more suitable for dealing with low-dimensional features. (2) The research mainly focuses on evaluating the classification performance of an SVM classifier trained by deep features of a specific layer of a single CNN model to find the best deep feature for classification.

With all the above, the objective of this research is to develop a new method for training the SVM classifier with CCA fused deep features for the identification of different varieties of grapes and thereby to overcome the problems with the current SVM+ deep feature methods. Inspired by Haghighat et al. [26] and Sun et al. [27], which demonstrated the outstanding performance by the fused feature with Canonical Correlation Analysis (CCA) fusion techniques, and considering working with small datasets, we hypothesize that a method can be developed that can fuse two kinds of deep features and reduce the dimensionality of fused deep features to improve the classification performance and reduce the training time. Thus, our new method would train the SVM classifier with CCA fused deep features for the identification of different varieties of grapes. By utilizing two CNNs to extract the deep features of the training set, and then CCA to fuse the obtained deep features and train the SVM with the fused features, we expect to deliver a better classifier for grape variety recognition. We also address the following specific hypotheses. (1) Compared with the deep feature extracted by a single CNN model, the fused deep feature from different networks can make the SVM classifier learn more features to improve the classification performance. (2) The CCA algorithm can eliminate redundant and invalid information when fusing different deep features, and significantly reduce the feature dimensionality to alleviate the problem that the model performance is affected when the SVM model learns high-dimensional features, and the reduction of feature dimensionality helps to speed up the model training speed.

The remainder of the article is organized as follows. In Section 2, the studied dataset and models and the proposed method are given. Then, in Section 3, the experiment results of different models are presented. Further, in Section 4, a comprehensive discussion based on the experiment results and the other studies are presented. Finally, a conclusion of the research is given in Section 5.

2. Materials and Methods

2.1. Dataset

One publicly available grapes image dataset named WGISD was adopted to evaluate the performance in this study, which can be downloaded from https://github.com/thsant/wgisd (accessed on 10 June 2021). The WGISD dataset consists of 300 images (2048 × 1365 pixels) with 5 different varieties (Chardonnay, Cabernet Franc, Cabernet Sauvignon, Sauvignon Blanc, and Syrah), and the detailed distribution of the dataset is as shown in Figure 1. In the experiment, the dataset was randomly split into 70:30 for training and testing, respectively.

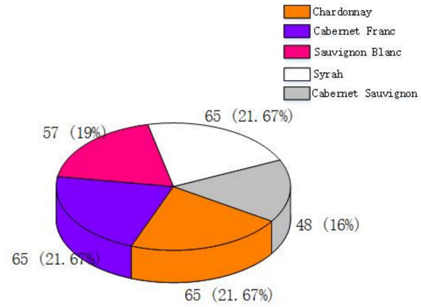

Figure 1. The varieties distribution of WGISD dataset.

2.2. Network Architecture and Deep Features Layers

In this research, the deep feature of three state-of-the-art CNN models, i.e., AlexNet [28], GoogLeNet [29], and ResNet [30], were adopted to evaluate the performance of the proposed method. The deep features are extracted from fully connected layer of a CNN model. Generally, a CNN may include several different fully connected layers (deep feature layers). For example, AlexNet consists of three deep feature layers, namely fc6, fc7, and fc8. Then, in this research, some typical deep feature layers were examined, and the detailed information of the selected layers was listed in Table 1.

Table 1. The deep feature layer and feature vector of the studied CNN.

CNN Model	Feature Layer	Feature Vector
AlexNet	fc6	4096
	fc7	4096
	fc8	1000
GoogLeNet	loss3-classifier	1000
ResNet18	Fc1000	1000
ResNet50	Fc1000	1000
ResNet101	Fc1000	1000

2.2.1. AlexNet

AlexNet was first proposed by Alex Krizhevsky et al. in the ImageNet competition and won first place in 2012. AlexNet is a deepening of the layers of the network based on LeNet, which enables it to learn richer and higher dimensional features. The proposal of AlexNet is the beginning of deep learning. It is a basic, simple, and effective CNN architecture, which is mainly composed of cascade stages, namely a convolutional layer, pooling layer, rectified linear unit (ReLU) layer, and fully connected layer. Specifically, the AlexNet consists of 5 convolutional layers, as shown in Figure 2, with pooling layers behind the first, second, third, and fourth layer, and 3 fully connected layers behind the fifth layer. The success of AlexNet can be attributed to some practical strategies, such as using ReLU nonlinear layers instead of sigmoid function as activation functions, using dropout to suppress overfitting, and using multi-GPU training. ReLU is a half-wave rectification function that can significantly accelerate the training phase and prevent overfitting. In addition, the dropout can be considered as a regularization to reduce the co-adaptation of neurons by setting the number of input neurons or hidden neurons to zero at random, which is usually used in the fully connected layers of AlexNet architecture. In this research, the deep features of fc6, fc7, and fc8 (which could be observed in Figure 2) of AlexNet were adopted to evaluate the performance of the SVM classifier.

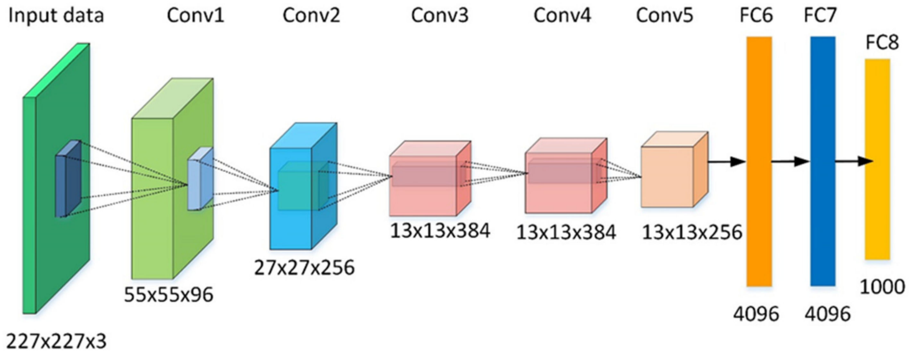

Figure 2. The architecture of AlexNet.

2.2.2. GoogLeNet

GoogLeNet, as shown in Figure 3, is a new deep learning structure proposed by Christian Szegedy in 2014. Before that AlexNet, VGG, and other networks obtained affordable performance by increasing the depth of the network (layers), but the increase in layers brought many negative effects, such as overfitting, gradient disappearance, and gradient explosion. To address this issue, a new module, Inception, was proposed by Szegedy et al. to construct the GoogLeNet network. The architecture of Inception is shown in Figure 4, which puts multiple convolutions or pooling operation together to form a single network unit. The proposal of Inception is to improve the performance from another perspective: it can use computing resources more efficiently and can extract more features with the same amount of calculation.

When designing a network that does not adopt the inception, we tend to use only one operation in a layer, such as convolution or pooling, and the size of the convolution kernel for the convolution operation is also fixed. However, in practical situations, for different sizes of images, different sizes of convolution kernels are needed to make the best performance, or, for the same image, different sizes of convolution kernels behave differently because they have different receptive fields. Therefore, we want to let the network choose by itself, and Inception can meet such needs. An Inception module provides multiple convolutional kernels in parallel, and the network chooses to use them by adjusting the parameters during training. In our research, the last fully connected layer (as shown in Figure 3) was chosen as the feature extraction layer. This layer is named "loss3-classifier" in the Deep Learning Toolbox of MATLAB and is usually chosen as the feature extraction layer of GoogLeNet for different applications [24,31,32].

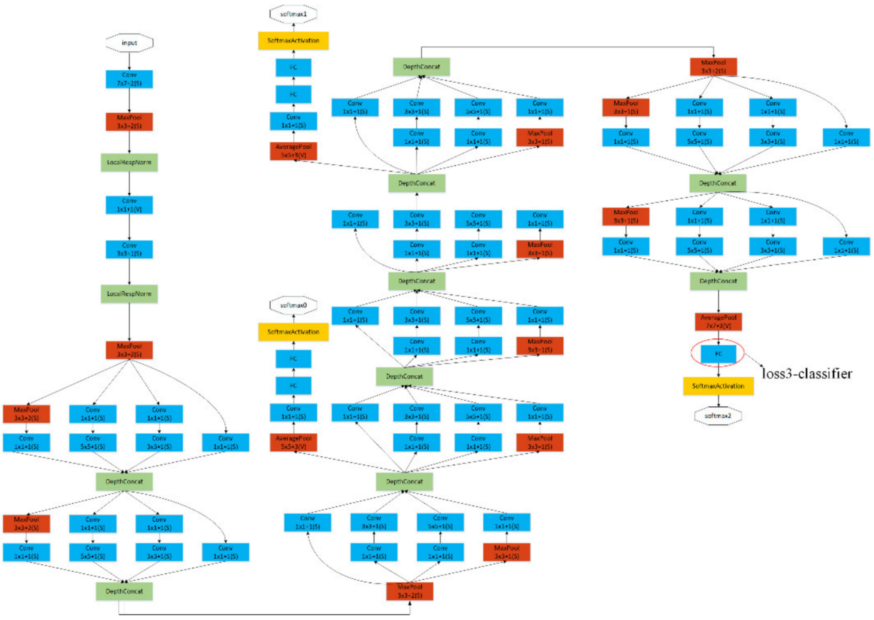

Figure 3. The architecture of GoogLeNet.

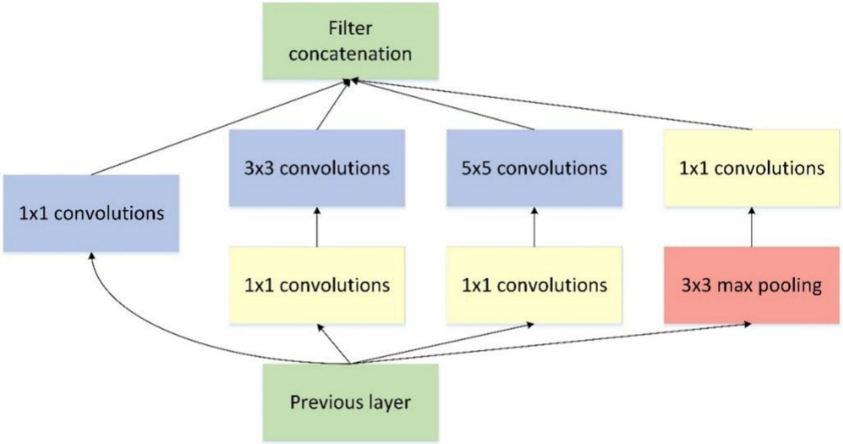

Figure 4. The Inception module proposed by GoogLeNet.

2.2.3. ResNet

As the network deepens, there is a decrease in the accuracy of the training set, and it is certain that this is not caused by overfitting (the accuracy should be high enough in the case of overfitting). To address this problem, He et al. proposed a new network named deep residual network (ResNet), which introduces a residual block structure, as shown in Figure 5, to solve the problem degradation in deep networks while allowing the network to be as deep as possible. The X Identity in Figure 5 is called “shortcut connection” and is the significant difference between the deep residual network and other networks. Two kinds of mapping were proposed by ResNet, the one is identity mapping and the other one is residual mapping. The output of a residual block is y = F(x) + x (do not consider nonlinear activation), and identity mapping refers to the input itself, which is x in the equation, while residual mapping refers to the F(x) part. If the network has reached the optimum, even if the network deepens, the residual mapping will be pushed to 0, leaving only identity mapping, so that the network is always in an optimum state and the performance of the network will not decrease as the layers increase.

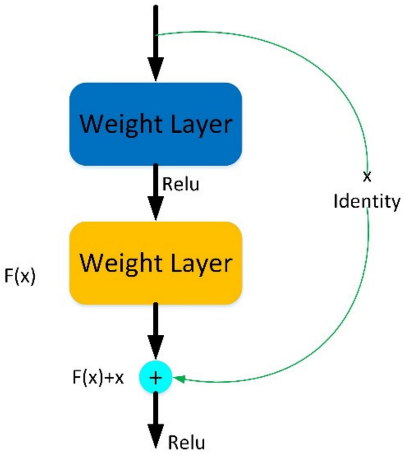

Figure 5. Residual block.

As shown in Equation (1), if the dimensions of x and F(x) are different, a linear projection W should be adopted on x such that the dimensions of x could match the dimensions of the F(x).

$$y = \mathrm{F}(\mathrm{x},\{\mathrm{W}_i\})+\mathrm{W}_x\mathrm{x}. \tag{1}$$

In our study, three widely used ResNet architectures, i.e., ResNet18, ResNet50, and ResNet101, were chosen as the deep feature extraction network. Similar to the "loss3-classifier", the fully connected layer before each ResNet was chosen as the deep feature extraction layer. These layers are named "Fc1000" in their respective networks and have also been wildly adopted for deep feature extraction for different applications [24,31,32].

2.3. Fusion of Deep Features by Canonical Correlation Analysis

In this research, two kinds of deep features extracted by the different networks of different deep feature layers were fused into a single feature vector by using a feature fusion technique based on Canonical Correlation Analysis (CCA) [27]. The fused feature is more discriminative than any of the input feature vectors. The Canonical Correlation Analysis (CCA) has been widely adopted to analyze associations between two sets of variables. The detailed mathematical derivation of CCA is shown in Appendix A.

As defined in [27], the deep features extracted by different CNN models could be fused by summation of the transformed features (canonical variates X^* and Y^*), and the fusion equation is shown in Equation (2).

$$\mathrm{Z} = \mathrm{X}^*+\mathrm{Y}^*= \mathrm{W}_x^{\mathrm{T}}\mathrm{X} + \mathrm{W}_y^{\mathrm{T}}\mathrm{Y} = \begin{pmatrix} \mathrm{W}_x \\ \mathrm{W}_y \end{pmatrix}^{\mathrm{T}} \begin{pmatrix} \mathrm{X} \\ \mathrm{Y} \end{pmatrix}, \tag{2}$$

where X^* and Y^* are the transformation of original extracted deep features X and Y by matrices W_x^T and W_y^T, respectively. Through matrix transformation, the matrices X and Y, whose dimensionalities are not necessarily equal, become equal. Further, the fused deep features Z can be obtained through numerical summation of corresponding positions of X^* and Y^*.

2.4. Proposed Methodology

The processing flow of the proposed method is illustrated in Figure 6.

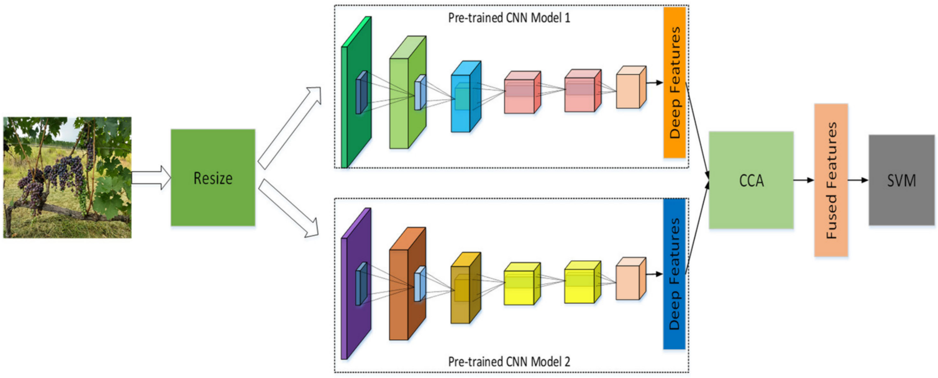

Figure 6. The processing flow of the proposed method.

Firstly, we resize the image to make it fit the input requirement of the CNN models. The CNN models used in this article were AlexNet, ResNet (18, 50, and 101), and GoogLeNet, and their input requirements are 227 × 227 × 3, 224 × 224 × 3, and 224 × 224 × 3, respectively.

Second, we input the image to the pre-trained CNN model to extract the deep features on specific layers of the CNN model. A typical CNN usually consists of two parts: convolutional base and classifier. The convolutional base is mainly used to automatically learn hierarchical features representations of the input image. Models trained with large datasets often have stronger feature learning abilities than trained with a small dataset. This means that a model pre-trained by a large dataset to extract deep features probably captures statistics of natural images much better than could have been learned from the WGISD dataset because of its limited size. Therefore, all the CNNs adopted for deep features extraction are pre-trained by ImageNet, which is a famous public dataset that contains 14 million images with 20 thousand classes.

Third, fuse the deep features extracted from different CNN models by Canonical Correlation Analysis (CCA) algorithm. The Canonical Correlation Discriminant Features is more discriminative than any of the input feature vectors. Compared with directly concatenating the deep features extracted from different CNN models, CCA can effectively eliminate the invalid features and reduce the feature dimensionality, which can avoid overfitting training and reduce the training time and computational resources of the SVM classifier.

Finally, the fused deep features were fed into a well-trained SVM classifier, then the SVM classifier could distinguish the grape variety and output the result. In the training stage, the function of "fit class error correcting output codes" (fitcecoc) was used, which could train a multi-class SVM classifier. The function of "fitcecoc" uses $K(K-1)/2$ binary SVM model with One-Vs-All coding design, which enhances the classification performance of the classifier.

2.5. Experiment Environment

The experiment computer is a Dell-T7920 (Austin, TX, USA) workstation running on Windows 10, and the hardware configuration is 2 Intel Xeon Gold 6248R CPUs, 64GB of RAM, and 2 Nvidia Quadro RTX 5000 graphics cards with 32 GB memory. The software environment is MATLAB 2020b (Netik, MA, USA), which supports the operation of CNN models such as AlexNet, GoogLeNet, and ResNet by installing the DeepLearning toolbox. In addition, the network parameters of AlexNet, GoogLeNet, and ResNet (18, 50, and 101) are pre-trained by ImageNet and have powerful feature extraction capability.

2.6. Performance Evaluation Metrics

To evaluate the performance of the proposed method, four metrics have been applied, i.e., accuracy [33], recall [33], precision [33], and F1 Score [33], which are most widely used to evaluate the performance of classification. The accuracy represents the ratio of the number of correctly classified samples to the total number of samples. The recall represents the ratio of the number of true positive samples to the positive samples. The precision represents the ratio of the number of true positive samples to the number of the samples predicted as positive, and the F1 Score is a harmonic metric, which takes into account both the recall and precision of the classification model. The equation of the adopted metrics is shown in Equations (3)–(6).

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{3}$$

$$\text{Rec}all = \frac{TP}{TP + FN} \tag{4}$$

$$\text{P}recision = \frac{TP}{TP + FP} \tag{5}$$

$$\text{F}1Score = 2 \times \frac{recall \times precision}{recall + precision} \tag{6}$$

where *TP* is true positive, *TN* is true negative, *FP* is false positive, and *FN* is false negative.

3. Results

In the experiment, in order to obtain more reliable experiment data, 10 independent runs of training and validation of each SVM classifier were made on the dataset, and the mean of results and deviation on the test set was adopted to represent its performance. Besides, since the standard deviations of the results of each model are very small, only the mean of the results were discussed.

3.1. Performance Analysis Based on Single Type of Deep Feature

The classification results and training time of the SVM classifier trained with a single type of deep feature are shown in Table 2. It is observed that the Fc1000 layer of ResNet101 obtained the best classification performance among all the examined deep layers, and the average accuracy, precision, recall, and F1 Score are 87.7%, 88.4%, 88%, and 88.2%, respectively. Besides, the fc6 of AlexNet can achieve better performance than fc7 and fc8, which was also indicated by other studies [24,34], reporting that the deep feature of fc6 is more distinguishable than that of fc7 and fc8. Then, in the following sections, the fc6 deep feature of AlexNet is considered only. Figure 7 shows the visualization of same sample with different deep feature. In addition, the training time of an SVM classifier with the fc6 layer of AlexNet is the longest, with an average of 2.5 s, while the average time of all other single type deep features was within 0.5 s. The dimension of the fc6 deep feature is 4096, and all the others are 1000, which may be the reason why the training time of fc6 is longer than that of other deep features.

Table 2. Performance metrics and training time of SVM classifier with a single deep feature (bold font shows the best performance).

Metrics	**AlexNet**			**GoogLeNet**	**ResNet18**	**ResNet50**	**ResNet101**
	fc6	**fc7**	**fc8**	**Loss3-Classifier**	**Fc1000**	**Fc1000**	**Fc1000**
Accuracy (%)	86.3 ± 1.1	81.5 ± 1.8	79.3 ± 0.9	72.5 ± 2.7	79 ± 2.4	83.4 ± 2	**87.7** ± 1.2
Precision (%)	87.8 ± 0.9	83.8 ± 1.4	81.4 ± 0.7	75.2 ± 2.8	80.9 ± 2.7	84.4 ± 2	**88.4** ± 1.4
Recall (%)	86.8 ± 1.1	82.3 ± 2	79.7 ± 1	73.3 ± 2.7	79.7 ± 2.2	84.1 ± 1.9	**88** ± 1.2
F1 Score (%)	87.3 ± 0.9	83.1 ± 1.7	80.6 ± 0.8	74.2 ± 2.7	80.3 ± 2.4	84.3 ± 1.9	**88.2** ± 1.2
Training time (S)	2.5 ± 0.39	0.4 ± 0.32	0.1 ± 0.04	0.09 ± 0.04	0.1 ± 0.02	0.1 ± 0.04	0.2 ± 0.27

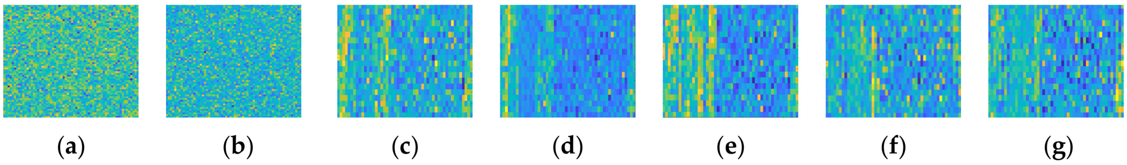

Figure 7. Visualization of different deep feature with the same sample: (**a**) fc6; (**b**) fc7; (**c**) fc8; (**d**) loss3-classifier; (**e**) Fc1000 of ResNet18; (**f**) Fc1000 of ResNet50; (**g**) Fc1000 of ResNet101.

3.2. Performance Analysis Based on CCA Fused Deep Features

Table 3 shows the performance of the SVM classifier trained by CCA fused deep features extracted from different CNNs on the test set. It can be observed that the combination of fc6 and ResNet50 (Fc1000) achieves the best classification performance, and the accuracy, precision, recall, and mean F1 Score are 96.5%, 97.0%, 96.7%, and 96.8%, respectively. The combination of GoogLeNet (loss3-classifier) and ResNet18 (Fc1000) obtained the worst performance, and the mean accuracy, precision, recall, and F1 Score are 90.6%, 91.1%, 91%, and 91.1%, respectively. For all the fused deep features, the dimensionality is 211 and the training times are usually within 0.1 s, which seems to verify the speculation about the relationship between training time and feature dimensionality in the previous section.

Table 3. Performance metrics and training time of SVM classifier with fused deep features (bold font shows the best performance).

Fused Features	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)	Train Time (S)	Dimension
fc6 + GoogleNet	94.5 ± 2.4	95.2 ± 2.1	94.7 ± 2.3	95.0 ± 2.2	0.07 ± 0.02	211
fc6 + ResNet18	92 ± 1.6	92.5 ± 1.3	92.2 ± 1.8	92.4 ± 1.5	0.06 ± 0.001	211
fc6 + ResNet50	**96.5** ± 1.4	**97.0** ± 1.2	**96.7** ± 1.5	**96.8** ± 1.3	0.06 ± 0.007	211
fc6 + ResNet101	96.1 ± 1	96.6 ± 0.9	96.2 ± 0.9	96.4 ± 0.9	0.07 ± 0.02	211
GoogleNet + ResNet18	90.6 ± 0.9	91.1 ± 0.7	91 ± 1	91.1 ± 0.8	0.07 ± 0.04	211
GoogleNet + ResNet50	92 ± 0.8	92.8 ± 0.5	92.1 ± 0.9	92.4 ± 0.6	0.09 ± 0.03	211
GoogleNet + ResNet101	95.2 ± 1.2	95.7 ± 1.1	95.4 ± 1	95.5 ± 1.1	0.05 ± 0.006	211
ResNet18 + ResNet50	92.2 ± 1.2	92.9 ± 1.3	92.3 ± 1.5	92.6 ± 1.4	0.07 ± 0.005	211
ResNet18 + ResNet101	92.7 ± 1	93.1 ± 0.8	92.9 ± 0.8	93.0 ± 0.7	0.07 ± 0.02	211
ResNet50 + ResNet101	96.1 ± 1	96.4 ± 1	96.2 ± 0.9	96.3 ± 1	0.18 ± 0.008	211

Table 4 shows the performance comparison between the proposed fused deep features and a single feature. The first row in the table is the best performance obtained by the fused deep features, while the second row is the worst performance, the third row is the best performance obtained by a single deep feature, and the fourth and fifth are the components of the fused deep features in the first row. Since the calculation of F1 Score combines precision and recall, we analyze the performance differences with mean F1 Score in this section.

Table 4. Performance comparison between the fused deep features and a single feature. (Bold font shows the best performance).

Fused Features	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)	Fusion Time (S)	Train Time (S)	Dimension
fc6 + ResNet50	**96.5** ± 1.4	**97.0** ± 1.2	**96.7** ± 1.5	**96.8** ± 1.3	0.05 ± 0.008	0.06 ± 0.007	211
GoogleNet + ResNet18	90.6 ± 0.9	91.1 ± 0.7	91 ± 1	91.1 ± 0.8	0.03 ± 0.002	0.07 ± 0.04	211
ResNet101	87.7 ± 1.2	88.4 ± 1.4	88 ± 1.2	88.2 ± 1.2	-	0.2 ± 0.27	1000
fc6	86.3 ± 1.1	87.8 ± 0.9	86.8 ± 1.1	87.3 ± 0.9	-	2.5 ± 0.39	4096
ResNet50	83.4 ± 2	84.4 ± 2	84.1 ± 1.9	84.3 ± 1.9	-	0.1 ± 0.04	1000

The best classification performance is obtained by the fusion of deep feature extracted from fc6 of AlexNet and ResNet50 (Fc1000), and the mean F1 Score is 96.8%, which is 8.6% higher than the best single deep feature. Further, the mean F1 Score of the worst fused deep features reaches 91.1%, which is also 2.9% higher than the best single feature. For fc6 and ResNet50 (Fc1000), the mean F1 Score of the fused deep feature is 9.5% and 12.5% higher than that of the single deep feature before the fusion, which indicates that the method proposed in this paper could improve the classification performance significantly. In addition, the training time plus feature fusion time of the proposed method is not more than the training time of a single feature, indicating that the proposed method is helpful to shorten the model training time.

Although the proposed method achieved satisfactory performance on the entire test set, the identification performance of the classifier on each variety of grape in the dataset is not clear. We further obtained the confusion matrix of the classifier's (fc6 deep feature + Fc100 of ResNet50) 10 independent runs on the test set, as shown in Table 5. The confusion matrix is a tool to evaluate the performance of supervised learning classification algorithm. In our research, each row of the confusion matrix represents the actual variety of grapes while each column represents the predicted variety, and the value on the diagonal represents the correct classification of each variety. It can be intuitively observed that the classifier achieved the best classification effect for Syrah. All 140 samples of Syrah were correctly identified. Following is Cabernet Sauvignon, for which all 164 samples were also correctly identified, but some other varieties were misidentified as Cabernet Sauvignon. However, the classification effect on the other three varieties cannot be directly observed from the confusion matrix, so we calculate the F1 Score of the classifier for each variety. The mean F1 Score of Syrah, Cabernet Sauvignon, Cabernet Franc, Sauvignon Blanc, and Chardonnay are 1, 0.982, 0.9738, 0.9399, and 0.9394, respectively. Therefore, we can obtain the actual ranking of the classification effect on each variety of grape by the classifier: Syrah > Cabernet Sauvignon > Cabernet Franc > Sauvignon Blanc > Chardonnay. Furthermore, through the intuitive observation of the test set, we seem to find that the closer the grape clusters to the camera, the better the classification performance. This gives us some inspiration that the camera should be as close to the grapevine as possible to capture the more clear clusters of grapes to improve the identification performance in practice.

Table 5. The confusion matrix of the 10 independent runs on the test set with fc6 + Fc1000 (ResNet50) fused deep features.

Grape Varieties	Cabernet Franc	Sauvignon Blanc	Chardonnay	Cabernet Sauvignon	Syrah
Cabernet Franc	187	2	0	5	0
Sauvignon Blanc	0	180	12	1	0
Chardonnay	3	8	178	0	0
Cabernet Sauvignon	0	0	0	164	0
Syrah	0	0	0	0	140

3.3. Performance Comparison with Using Deep Network Directly

Further, the performance of the proposed method (fused deep features plus SVM) was compared with the performance obtained by using CNNs directly. Commonly, the number of classification classes of the original CNN is often inconsistent with the number of classification classes we study, which makes it impossible for us to directly apply the original network to our applications. Therefore, the meaning of directly using CNN in this article essentially refers to directly using the classification output of a CNN. In order to make a CNN suitable for our study, the parameter of the final "softmax" (the classifier in CNN) should be adjusted to be 5 (number of grape varieties in the WGISD). Table 6 shows the training parameters for the examined CNN models. In the experiment, "stochastic gradient descent with momentum" (sgdm) was selected as the optimization algorithm (solver), and the parameters of "MiniBatchSize" (the number of samples utilized in one iteration), "InitialLearnRate", and "MaxEpochs" (the number of passes of the entire training dataset) were 20, 1e-3, and 50, respectively. Moreover, because the workstation has two GPU, the parameter of "ExecutionEnvironment" was set as "multi-gpu" to speed up the training.

Table 6. Training parameters of the examined CNN models.

Parameters	Value
solver	sgdm
MiniBatchSize	20
InitialLearnRate	1e-3
MaxEpochs	30
ExecutionEnvironment	multi-gpu

The performance comparison between the method proposed in this article and using the CNN directly is obtained in Table 7. On the one hand, from the perspective of classification performance, the mean F1 Scores of the AlexNet, GoogLeNet, ResNet18, ResNet50, and ResNet101 are 87.0%, 94.2%, 90.7%, 88.5%, and 82.3%, respectively. The mean F1 Scores of the proposed method are 9.8%, 2.6%, 6.1%, 8.3%, and 14.5% higher than AlexNet, GoogLeNet, ResNet18, ResNet50, and ResNet10, respectively, which indicate that, for the identification of grape varieties, the proposed method has better classification performance than using CNN directly. However, the CNN often contains a large number of parameters, which requires a large number of samples to train and fine tune the parameters. Therefore, we believe that the small number of samples of the public dataset studied in this paper is the reason why it is difficult to obtain better classification performance by using the CNN directly. It is not recommended to train a deep network with small datasets to avoid the problem of overfitting. However, the method proposed in this paper can still achieve good classification performance (the mean F1 Score is as high as 96.8%) with the small dataset, showing obvious advantages. On the other hand, the training time also improved with respect to the experimental environment of our study. To sum up, the results indicate that the method proposed in this paper have advantages both in identification performance and training time compared with using CNN directly.

Table 7. Performance comparison between the proposed method and using CNN directly (bold font shows the best performance).

Method	Accuracy	Precision	Recall	F1 Score	Train Time
AlexNet	85.7 ± 1.3	88.4 ± 1.2	85.7 ± 1.1	87.0 ± 0.9	4228 ± 53
GoogLeNet	93.8 ± 0.8	94.6 ± 0.9	94.0 ± 1.1	94.2 ± 1.0	4222 ± 47
ResNet18	89.8 ± 1.3	91.6 ± 1.2	90 ± 1.1	90.7 ± 1.2	4247 ± 38
ResNet50	87.7 ± 1.1	89.1 ± 1.2	88.0 ± 1.4	88.5 ± 1.3	4310 ± 51
ResNet101	81.6 ± 1.4	83.0 ± 1.3	81.7 ± 1.5	82.3 ± 1.3	4377 ± 63
Fused deep features (proposed by this study)	**96.5** ± 1.4	**97.0** ± 1.2	**96.7** ± 1.5	**96.8** ± 1.3	0.06 ± 0.007

4. Discussion

In this study, a new method based on the CCA feature fusion algorithm plus SVM was proposed. Donahue et al. [35] and Razavian et al. [36] demonstrated the outstanding performance of the pre-trained CNN features in various recognition tasks. Although the model trained from scratch may achieve comparative or better performance, it will be more time-consuming. Besides, one potential limitation is that the amount of dataset used in our research is not particularly large, so training the network directly from scratch may cause over-fitting problems, which may cause degrading of the performance. Therefore, the pre-trained CNN models were adopted to extract features. Haghighat et al. [26] and Sun et al. [27] demonstrated an outstanding performance by the fused features with CCA fusion techniques. In our experiment, we also tried the fusion methods of direct concatenation and concatenation based on PCA, but there is a certain performance margin compared with CCA. Therefore, the CCA was selected to fuse the deep feature, which can find the most relevant features based on the two sets of features, so that better features can be used to extract further features from the comparative poorer features to enhance the performance.

The given CCA scheme is different from other fusion techniques (e.g., ensemble learning and concatenation). To be specific, ensemble learning and concatenation does not consider the correlations between the two input items from the perspective of features and may lose some valuable information. The experiment results on the public WGISD show that the proposed method could achieve satisfactory performance. However, there are many fusion algorithms in practice. In the future, more studies should be focused on selecting a more effective feature fusion algorithm to further improve the performance.

Furthermore, we compared our experimental results with some of the existing literature on grape varieties identification. Bogdan et al. cascaded the output of ResNet50 into a multi-layer perceptron, which can improve the classification accuracy, but this method needs to train two models. The evaluated dataset literature [20] is the same as ours, but the background was removed and each cluster of grapes was extracted separately for training and testing and an accuracy of 99% was obtained. However, in [22], when the background was not removed (the dataset is exactly the same as ours), the accuracy dropped by 26%, from 99% to 63%. This phenomenon suggests that (1) compared with their method, our proposed method can achieve higher classification results, and That (2) the preprocessing, the removal of the background, also plays a key role in the classification effect of the model. Pereira et al. used AlexNet to identify grape varieties collected under natural conditions. Although it carries out a variety of preprocessing on the dataset (does not include the removal of background), the accuracy obtained is 77.3%, which is 19.2% lower than ours. However, when this preprocessing was adopted to another popular Flavia leaf dataset, an accuracy of 89.75% was achieved, which indicates that a different dataset also has a large difference in results. This shows that although our approach has great advantages over theirs, the impact of different datasets on the results cannot be ignored. El-Mashharawi et al. also adopted a CNN for grape varieties identification, and an accuracy of 100% was achieved on a public dataset from Kaggle. Each image on the dataset only contains one grape grain and the background is pure white (the background pixels are all in white color). Based on the influence of background processing on classification performance in the above discussion of literature [20], we believe that the clear background is also an important reason for such high accuracy. In summary, the method proposed in this study could obtain satisfactory performance for grape varieties identification with the images collected under natural conditions. However, image preprocessing should be considered to improve the performance of our method. In addition, although the proposed method in this paper has made promising progress with the method of single deep feature + SVM, a satisfactory result was achieved on the WGISD dataset. As mentioned above, datasets also play an important role in classification performance. The method that we propose needs to be verified on different datasets to prove its versatility.

Agriculture is rapidly evolving towards a new paradigm—Agriculture 4.0, and digital technology, artificial intelligence (AI), and automation will play a particularly important role at this stage. Traditional manual machinery is gradually being replaced by smart machinery. To make the machinery smart like a human being as much as possible, an intelligent algorithm is necessary because different varieties of grapes have different harvesting time, nutrient requirements, and susceptibility to diseases and insect pests. With the help of the proposed algorithm, the smart machinery could carry out more scientific and reasonable operations for harvesting, fertilization, and management of different varieties of grapes, reducing input and improving grape quality and yield at the same time. In addition, the algorithm proposed in this paper is not only suitable for grapes but it can also be used for the identification of other fruits or vegetables for a more extensive application. However, there are still a lot of efforts that need to be made to enable our algorithm to be adopted in smart machineries. (1) The algorithm is an independent module, and when it is mounted on the smart machinery, it needs to consider the interconnection with the front-end image sensor and the back-end actuator. This way, only the proposed algorithm will have practical significance. (2) The image sensors are easily affected by the changeable field environment, and further affect the performance of the algorithm. More efforts should be adopted to deal with such problems.

5. Conclusions

In this research, a new method was proposed for the identification of different varieties of grapes based on small datasets. Our results showed that public WGISD datasets can be successfully used for identification of grape varieties and indicated that the fusion of two kinds of deep features by CCA can not only produce more distinguishable features for improving the classification performance, but can also eliminate redundant and invalid information, and thereby speed up the model training. Based on the proposed algorithm, smart machinery will have the potential of taking more targeted measures according to the different characteristics of different varieties of grapes, thus further improving their performance that involves grape varieties recognition. However, to apply more effectively the proposed algorithm to smart machinery, more images should be collected in the future for different varieties from different vineyards with various conditions to make the model more versatile. In addition, a complete software should be implemented, which includes not only the grape varieties identification module but also all the functional modules.

Author Contributions: Conceptualization, Y.P. and J.L.; methodology, Y.P.; software, Y.P. and S.Z.; writing—original draft preparation, Y.P.; writing—review and editing, Y.P.; supervision, J.L.; funding acquisition, J.L. All authors have read and agreed to the published version of the manuscript.

Funding: The research was funded by grants from the National Science Foundation of China (Grant No. 31971795), A Project Funded by the Priority Academic Program Development of Jiangsu Higher Education Institutions (No. PAPD-2018-87) and Project of Faculty of Agricultural Equipment of Jiangsu University (4111680002).

Institutional Review Board Statement: Not applicable.

Data Availability Statement: Not applicable.

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

Appendix A

Suppose that two different methods used to extract the p dimensional and q dimensional deep features of each sample, and two matrices, $X \in R^{p\times n}$ and $Y \in R^{q\times n}$, are obtained, where n is the number of samples (the number of samples trained in training phase, or the number of test samples in testing phase). In other words, a total of (p+q) dimensional features are extracted for each sample.

Let $S_{xx} = R^{p\times p}$ and $S_{yy} = R^{q\times q}$ denote the within-sets covariance matrices of X and Y, and $S_{xy} = R^{p\times q}$ denote the between-sets covariance matrix between X and Y. The matrix S shown below is the overall $(p+q)\times(p+q)$ covariance matrix, which contains all the information on associations between the pairs of deep features.

$$S = \begin{pmatrix} \mathrm{cov}(x) & \mathrm{cov}(x,y) \\ \mathrm{cov}(y,x) & \mathrm{cov}(y) \end{pmatrix} = \begin{pmatrix} S_{xx} & S_{xy} \\ S_{yx} & S_{yy} \end{pmatrix}. \tag{A1}$$

However, the correlation between these two sets of deep feature vectors may not follow a consistent pattern, and therefore, it is difficult to understand the relationship between these two sets of deep features from this matrix [37]. The aim of CCA is to find a linear transformation,$\mathrm{X}^*= \mathrm{W}_x^{\mathrm{T}}\mathrm{X}$ and $\mathrm{Y}^*= \mathrm{W}_y^{\mathrm{T}}\mathrm{Y}$, and to maximize the pair-wise correlation between the two datasets:

$$\mathrm{corr}(\mathrm{X}^*,\mathrm{Y}^*) = \frac{\mathrm{cov}(\mathrm{X}^*,\mathrm{Y}^*)}{\mathrm{var}(\mathrm{X}^*)\cdot\mathrm{var}(\mathrm{Y}^*)}, \tag{A2}$$

where $\text{cov}(\text{X}^*, \text{Y}^*) = \text{W}_x^{\text{T}}\text{S}_{xy}\text{W}_y$, $\text{var}(\text{X}^*) = \text{W}_x^{\text{T}}\text{S}_{xx}\text{W}_x$ and $\text{var}(\text{Y}^*) = \text{W}_y^{\text{T}}\text{S}_{yy}\text{W}_y$. The covariance between X^* and Y^* ($\text{X}^*, \text{Y}^*\ \hat{\text{I}}\text{R}^{\text{d}\times\text{n}}$ are known as canonical variables) is maximized by Lagrange multiplier method, and the constraint condition is $\text{var}(\text{X}^*) = \text{var}(\text{X}^*) = 1$. Further, the linear transformation matrix W_x and W_y can be obtained by solving the eigenvalue equation as below [37]:

$$\begin{cases} \text{S}_{xx}^{-1}\text{S}_{xy}\text{S}_{yy}^{-1}\text{S}_{yx}\hat{\text{W}}_x = \Lambda^2\hat{\text{W}}_x \\ \text{S}_{yy}^{-1}\text{S}_{yx}\text{S}_{xx}^{-1}\text{S}_{xy}\hat{\text{W}}_y = \Lambda^2\hat{\text{W}}_y \end{cases}, \tag{A3}$$

where $\hat{\text{W}}_x$ and $\hat{\text{W}}_y$ are the eigenvectors and Λ^2 is a diagonal matrix of eigenvalues or squares of the canonical correlations.

The number of non-zero eigenvalues of each equation is $d = rank(S_{xy}) \leq \min(n, p, q)$, further arranged in descending order, $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d$. The transformation matrix W_x and W_y are composed of eigenvectors corresponding to sorted non-zero eigenvalues. For the transformed data, the form of the sample covariance matrix defined in Equation (7) is as follows:

$$S^* = \left(\begin{array}{cccc|cccc} 1 & 0 & \cdots & 0 & \lambda_1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 & 0 & \lambda_2 & \cdots & 0 \\ \vdots & & \ddots & & \vdots & & \ddots & \\ 0 & 0 & \cdots & 1 & 0 & 0 & \cdots & \lambda_d \\ \lambda_1 & 0 & \cdots & 0 & 1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 & 0 & 1 & \cdots & 0 \\ \vdots & & \ddots & & \vdots & & \ddots & \\ 0 & 0 & \cdots & \lambda_d & 0 & 0 & \cdots & 1 \end{array}\right). \tag{A4}$$

As shown in the above matrix, the upper left and lower right identity matrices indicate that the canonical variates are uncorrelated within each dataset, and canonical variates have non-zero correlation only on their corresponding indices.

References

1. Pereira, C.S.; Morais, R.; Reis, M.J.C.S. Deep learning techniques for grape plant species identification in natural images. *Sensors* **2019**, *19*, 4850. [CrossRef]
2. Botterill, T.; Paulin, S.; Green, R.; Williams, S.; Lin, J.; Saxton, V.; Mills, S.; Chen, X.; Corbett-Davies, S. A robot system for pruning grape vines. *J. Field Robot.* **2017**, *34*, 1100–1122. [CrossRef]
3. Monta, M.; Kondo, N.; Shibano, Y. Agricultural robot in grape production system. In Proceedings of the 1995 IEEE International Conference on Robotics and Automation, Nagoya, Japan, 21–25 May 1995; pp. 2504–2509.
4. Ogawa, Y.; Kondo, N.; Monta, M.; Shibusawa, S. Spraying robot for grape production. In *Springer Tracts in Advanced Robotics*; Springer: Berlin/Heidelberg, Germany, 2003; pp. 539–548.
5. Kondo, N. Study on grape harvesting robot. *IFAC Proc.* **1991**, *24*, 243–246. [CrossRef]
6. Versari, A.; Laurie, V.F.; Ricci, A.; Laghi, L.; Parpinello, G.P. Progress in authentication, typification and traceability of grapes and wines by chemometric approaches. *Food Res. Int.* **2014**, *60*, 2–18. [CrossRef]
7. Duhamel, N.; Slaghenaufi, D.; Pilkington, L.I.; Herbst-Johnstone, M.; Larcher, R.; Barker, D.; Fedrizzi, B. Facile gas chromatography–tandem mass spectrometry stable isotope dilution method for the quantification of sesquiterpenes in grape. *J. Chromatogr. A* **2018**, *1537*, 91–98. [CrossRef] [PubMed]
8. Karimali, D.; Kosma, I.; Badeka, A. Varietal classification of red wine samples from four native Greek grape varieties based on volatile compound analysis, color parameters and phenolic composition. *Eur. Food Res. Technol.* **2020**, *246*, 41–53. [CrossRef]
9. Pérez-Navarro, J.; Da Ros, A.; Masuero, D.; Izquierdo-Cañas, P.M.; Hermosín-Gutiérrez, I.; Gómez-Alonso, S.; Mattivi, F.; Vrhovsek, U. LC-MS/MS analysis of free fatty acid composition and other lipids in skins and seeds of Vitis vinifera grape cultivars. *Food Res. Int.* **2019**, *125*, 108556. [CrossRef]
10. Kyraleou, M.; Kallithraka, S.; Gkanidi, E.; Koundouras, S.; Mannion, D.T.; Kilcawley, K.N. Discrimination of five Greek red grape varieties according to the anthocyanin and proanthocyanidin profiles of their skins and seeds. *J. Food Compos. Anal.* **2020**, *92*, 103547. [CrossRef]
11. Benbarrad, T.; Salhaoui, M.; Kenitar, S.B.; Arioua, M. Intelligent machine vision model for defective product inspection based on machine learning. *J. Sens. Actuator Netw.* **2021**, *10*, 7. [CrossRef]
12. Penumuru, D.P.; Muthuswamy, S.; Karumbu, P. Identification and classification of materials using machine vision and machine learning in the context of industry 4.0. *J. Intell. Manuf.* **2020**, *31*, 1229–1241. [CrossRef]

13. Mavridou, E.; Vrochidou, E.; Papakostas, G.A.; Pachidis, T.; Kaburlasos, V.G. Machine vision systems in precision agriculture for crop farming. *J. Imaging* **2019**, *5*, 89. [CrossRef]
14. Radcliffe, J.; Cox, J.; Bulanon, D.M. Machine vision for orchard navigation. *Comput. Ind.* **2018**, *98*, 165–171. [CrossRef]
15. Monkman, G.G.; Hyder, K.; Kaiser, M.J.; Vidal, F.P. Using machine vision to estimate fish length from images using regional Convolutional Neural Networks. *Methods Ecol. Evol.* **2019**, *10*, 2045–2056. [CrossRef]
16. Sung, H.-J.; Park, M.-K.; Choi, J.W. Systems. Automatic grader for flatfishes using machine vision. *Int. J. Control. Autom. Syst.* **2020**, *18*, 3073–3082.
17. Nasirahmadi, A.; Ashtiani, S.-H.M. Bag-of-Feature model for sweet and bitter almond classification. *Biosyst. Eng.* **2017**, *156*, 51–60. [CrossRef]
18. Bhargava, A.; Bansal, A. Classification and grading of multiple varieties of apple fruit. *Food Anal. Methods* **2021**, *14*, 1–10. [CrossRef]
19. Ponce, J.M.; Aquino, A.; Andújar, J.M. Olive-fruit variety classification by means of image processing and Convolutional Neural Networks. *IEEE Access* **2019**, *7*, 147629–147641. [CrossRef]
20. Franczyk, B.; Hernes, M.; Kozierkiewicz, A.; Kozina, A.; Pietranik, M.; Roemer, I.; Schieck, M. Deep learning for grape variety recognition. *Procedia Comput. Sci.* **2020**, *176*, 1211–1220. [CrossRef]
21. El-Mashharawi, H.Q.; Abu-Naser, S.S.; Alshawwa, I.A.; Elkahlout, M. Grape type classification using deep learning. *Int. J. Acad. Eng. Res.* **2020**, *3*, 14–45.
22. Acortes, C.; Vapnik, V. Support vector networks. Machine Learning. *Mach. Learn.* **1995**, *20*, 273–297. [CrossRef]
23. Aizerman, M.A. Theoretical foundations of the potential function method in pattern recognition learning. *Autom. Remote. Control.* **1964**, *25*, 821–837.
24. Sethy, P.K.; Barpanda, N.K.; Rath, A.K.; Behera, S.K. Deep feature based rice leaf disease identification using Support Vector Machine. *Comput. Electron. Agric.* **2020**, *175*, 105527. [CrossRef]
25. Sethy, P.K.; Barpanda, N.K.; Rath, A.K.; Behera, S.K. Nitrogen deficiency prediction of rice crop based on Convolutional Neural Network. *J. Ambient. Intell. Humaniz. Comput.* **2020**, *11*, 5703–5711. [CrossRef]
26. Haghighat, M.; Abdel-Mottaleb, M.; Alhalabi, W. Fully automatic face normalization and single sample face recognition in unconstrained environments. *Expert Syst. Appl.* **2016**, *47*, 23–34. [CrossRef]
27. Sun, Q.-S.; Zeng, S.-G.; Liu, Y.; Heng, P.-A.; Xia, D.-S. A new method of feature fusion and its application in image recognition. *Pattern Recognit.* **2005**, *38*, 2437–2448. [CrossRef]
28. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep Convolutional Neural Networks. *Adv. Neural Inf. Process. Syst.* **2012**, *25*, 1097–1105. [CrossRef]
29. 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.
30. 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.
31. Sethy, P.K.; Behera, S.K.; Ratha, P.k.; Biswas, P. Detection of coronavirus disease (covid-19) based on deep features and Support Vector Machine. *Int. J. Math. Eng. Manag. Sci.* **2020**, *5*, 643–651. [CrossRef]
32. Kadhim, M.A.; Abed, M.H. Convolutional Neural Network for satellite image classification. In *Studies in Computational Intelligence*; Springer: Cham, Switzerland, 2019; pp. 165–178.
33. Castelli, M.; Vanneschi, L.; Largo, Á.R. Supervised learning: Classification. In *Encyclopedia of Bioinformatics and Computational Biology*; Elsevier: Amsterdam, The Netherlands, 2018; Volume 1, pp. 342–349.
34. Chan, G.C.; Muhammad, A.; Shah, S.A.; Tang, T.B.; Lu, C.-K.; Meriaudeau, F. Transfer learning for diabetic macular edema (DME) detection on optical coherence tomography (OCT) images. In Proceedings of the 2017 IEEE International Conference on Signal and Image Processing Applications (ICSIPA), Kuching, Malaysia, 12–14 September 2017; pp. 493–496.
35. Donahue, J.; Jia, Y.; Vinyals, O.; Hoffman, J.; Zhang, N.; Tzeng, E.; Darrell, T. Decaf: A deep convolutional activation feature for generic visual recognition. In Proceedings of the 31th International Conference on Machine Learning, Beijing, China, 21–26 June 2014; pp. 647–655.
36. Sharif Razavian, A.; Azizpour, H.; Sullivan, J.; Carlsson, S. CNN features off-the-shelf: an astounding baseline for recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Columbus, OH, USA, 23–28 June 2014; pp. 806–813.
37. Krzanowski, W. *Principles of Multivariate Analysis*; OUP Oxford: Oxford, UK, 2000; Volume 23.