Using Deep Learning Ensemble Models for Grapevine Leaf Image Classification

[1]DR. P. Ramasubramanian, [2] T.Pranathi

[1] Professor, Megha Institute of Engineering & Technology for Women, Ghatkesar.
[2] MCA Student, Megha Institute of Engineering & Technology for Women, Ghatkesar.

Abstract

The nutritional value of grapes is significant, and they have many practical applications beyond just eating and fermenting. The ability to distinguish between different grape varieties according to the shape of their leaves is crucial for the propagation of grape varieties and the study of grape evolution. The article's study object is the mature leaves of several grape types. Collecting and preprocessing leaf pictures is the first step in training and fine-tuning five pre-trained deep learning models: VGG19, VIT, Inception ResnetV2, DenseNet201, and ResneXt. Lastly, the five models' predictions are combined using two voting ensemble machine learning models. Using a decision-making process based on soft voting, the ensemble classifier achieves the greatest accuracy of 98.1%.

Keywords-

Deep learning ensemble model for grapevine variety identification Contrasting Approaches to Voting

I. INTRODUCTION

Identifying grape varieties is crucial for spreading awareness of grape research and promoting this cash crop as the grape market economy grows. The grape variety identification study often uses the leaves as the object of identification. To achieve grape variety recognition using grape leaves, one must preprocess the image of the leaf. Then, features can be extracted from either deep learning or artificial design. Lastly, a recognition model can be built using a classifier and the extracted feature vectors as parameter inputs. Because it relies on human intervention to extract design features, it is tedious, time-consuming, and susceptible to human error. Machine vision, natural language processing, and other areas have made use of deep learning-based feature extraction. One kind of machine learning method, known as ensemble learning, trains several learners and then uses them together. The prediction outcomes obtained by this algorithm type are often superior than those of a single learner in real-world scenarios. To categorize photos of grapevine leaves, this research used the ensemble learning technique. We must first argue for the training sample set if we want to train our deep learning ensemble model accurately. Following the use of augmentation techniques, the training set was enlarged to 2800. As a second step, we classified grapevine leaves using five different models: VGG19[1], VIT[2], Inception ResnetV2[3], DenseNet201[4], and ResneXt [5]. The third point is that hard voting and soft voting were both used. Here is how the remainder of this paper is structured. Part II discusses the relevant literature. Our study approach, picture preprocessing method, deep feature extraction methods, voting strategies, and results comparison are detailed in Section III. The article is concluded and future work is addressed in Section IV.

II. RELATED WORKS

Classifying photos of grapevine leaves using machine learning techniques has been the subject of much study. A variety of machine learning techniques have been used, including Support Vector Machine, Logistic Regression, Bayesian Belief Network, and more. In order to categorize grapevine leaves, Hunar A. Ahmed modified DenseNet201. With a maximum accuracy of 98%, DenseNet 201 produced the best possible results [6]. In order to classify grapevine leaves, M. Koklu suggested a CNN-SVM research using chosen deep characteristics. In order to categorize grapevine leaves, they used SVM kernels and a pre-trained MobileNetv2 Logits layer for feature extraction. They found that their approach had a 97.6% success rate in categorization [7]. Images of grapevine leaves were semantically segmented for phenotyping purposes by Tamvakis Petros using U-Net architecture[8]. Variegated leaf veins and blade features were their primary areas of interest. They employed three distinct supervised learning methods: architectural parameterization, design-and-train from scratch, and transfer learning. In order to identify illnesses in grape leaves, Chen Yiping suggested a model based on deep learning that consists of three stages. They identified lesions on grape leaves using ResNet, augmented the data using generative adversarial networks, and used Faster R-CNN for lesion detection. Their suggested model had strong generalizability, as shown by their experimental findings [9]. Bingpiao Liu put up the YOLOX-RA grape detection model to tackle the issue of grape identification in unstructured situations. This model is capable of precisely and swiftly identifying clusters of densely growing grapes as well as grapes that are obstructed. Their model was able to attain a recognition speed of 84.88 FPS and a mAP of 88.75% [10]. For the purpose of real-time grape bunch recognition, Sozzi, M. used six variants of the YOLO object detection method. According to their findings, YOLOv5x and YOLOv4 achieved F1-scores of 0.76 and 0.77, respectively, [11].

III. METHODGOLOGY

Our research suggests using an ensemble approach to identify different species of grape leaves. Data preparation is the first step in getting the data ready for analysis. Afterwards, the dataset was trained using five classifiers: VGG19, VIT, Perception ResnetV2, DenseNet201, and ResneXt. When learning takes place, the output from all of the classifiers is combined. We use both hard voting and soft voting as integration tactics in our strategy. The ensemble method's process is shown in Fig. 1. Here you may find the dataset description, methods for picture preparation, algorithms for classification, and procedures for voting.

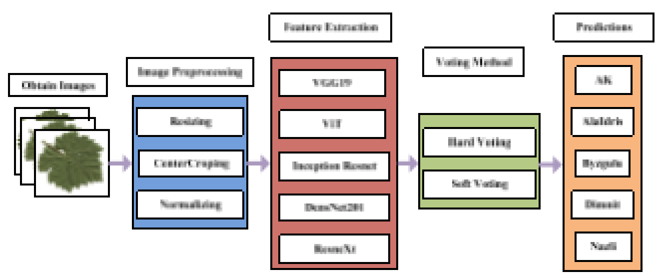

Fig. 1. Workflow of the methodology

Part A. The data collection There are a total of 500 leaf samples in the grapevine leaf dataset, which includes 5 species with 100 samples per class [7]. The five groups in question are AK, AlaIdris, Buzgulu, Dimnit, and Nazli. Specific information about each category's features differs from those of other categories. Leaf size, shape, and texture are all part of the feature data. We will train and identify five grape leaves, as shown in Fig. 2.

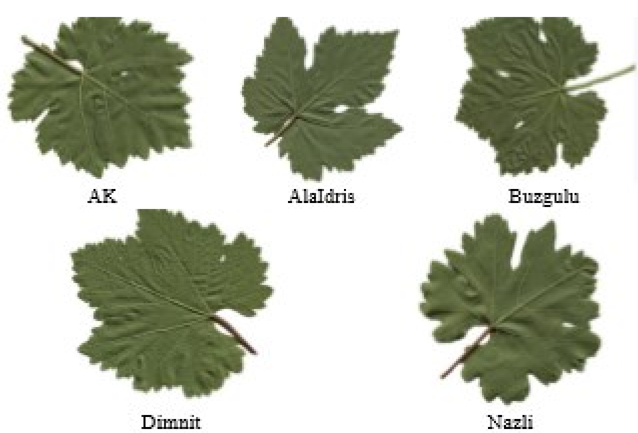

Fig. 2. Five grapevine leaf categories

Section B: Picture Preparation In order to reduce the processing cost, we downsize the picture from its original dimensions of 512 by 512 pixels to 256 x 256 pixels. In addition, the picture is cropped to 224 by 224 in the middle. Using a mean of 0.4850 and a standard deviation of 0.2240 and 0.2250, we normalize the picture. With p=0.5, scale=(0.02,0.32), and ratio=(0.3,3.2), we also use random erasing. Furthermore, the train set is randomly supplemented with Gaussian noise. The flowchart of image processing is shown in Figure 3.

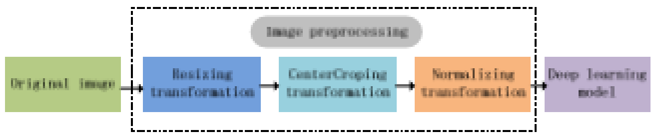

Fig. 3. Schematic diagram of image processing

B. Extracting deep features Grapevine leaf feature extraction is carried out in this work using five pre-trained models: VGG19, VIT, Inception ResnetV2, DenseNet201, and ResneXt. The first step is to fine-tune all of the chosen models. There are a total of nineteen hidden layers in VGG19, including sixteen convolutional layers and three fully linked layers [1]. While the original VGG19 model output has a thousand classes, our investigation only requires five. Hence, the layer's out channel number is modified to 5 from 1000. The vit_base_patch16_224 (ViTB/16 model) was used for testing and training purposes in the ViT model [2]. The MLP Head, Transformer Encoder, and Linear Projection of Flattened Patches are the three main components of the ViT-B/16 model. The dimensions of the input picture used by the ViT-B/16 model are 224 × 224 × 3. Its dimensions are 16 x 16 × 3. There are 12 transformer encoder blocks, each with a 768-by-dimension, and 12 heads used in Multi Head Attention. The first film Thirdly, we have ResnetV2, a DL model with 825 layers and 1000 item category classification capabilities [3]. The final complete connection layer's output channel number has to be set to 5. At 201 layers deep, Densnet201 is one CNN [4]. The Densnet201 model combines all of the feed-forward layers into one. In order to decrease computational complexity, ResNeXt employs grouping convolution, which entails dividing the feature graph into several groups and then convolutioning each group of feature graphs independently [5]. Techniques for Casting Votes (D) In order to arrive at the best possible result, the voting classifier merges the basis models. One possible mathematical representation of the ensemble classifier prediction is (1):

$$\hat{y} = \arg\max_{i} \sum_{j=1}^{m} w_j \, \mathcal{X}_A(C_j(x) = i) \qquad (1)$$

Cj is the classifier and wj is the weight linked to the classifier's prediction in the previous equation. This research presents two different methods of voting. You have two options: hard voting and soft voting. Hard voting relies on the minority caving in to the majority in order to get a final decision. A soft vote is one in which all classifier probabilities are added together. Two voting methods are compared. Figure 4 showed the algorithm for the suggested approach [12].

```
Algorithm 1
1: procedure  Preprocess(grapevine_leaf_data)
2:return grapevine_leaf_data['AK','AlaIdris','Buzgulu','Dimnit','Nazli']

3:procedure split_data(grapevine_leaf_data)
4:    Training_data, Testing_data=split(grapevine_leaf_data)
5:    returnTraining_data, Testing_data

6:D1=VGG(Training_data, Testing_data)
7:D2= ViT(Training_data, Testing_data)
8:D3=Inception Resnet(Training_data, Testing_data)
9: D4= DenseNet (Training_data, Testing_data)
10: D5= ResneXt (Training_data, Testing_data)

11:procedureensemble_model(Training_data, Testing_data))
12:    soft_voting_classifier=concatenate(D1,D2,D3,D4,D5)
13:    soft_voting_classifier.fit(Training_data)
14:    soft_predictions= hard_voting_classifier.predict(Testing_data)
15:    hard_voting_classifier=concatenate(D1,D2,D3,D4,D5)
16:    hard_voting_classifier.fit(Training_data)
17:    hard_predictions= hard_voting_classifier.predict(Testing_data)
```

Fig. 4. Algorithm for proposed ensemble soft voting Classifier

E. Outcome We split the dataset in three parts: "train," "test," and "validation," with each set comprising 80%, 10%, and 10% of the total. Table 1 displays the method's training hyperparameters. We set EPOCHS to 9 since the machine's performance was restricted. Raising the value of EPOCHS actually makes recognition more precise.

Table 1. TRAINING HYPERPARAMETER SETTING

Hyperparameter	Value
Batch_Size	64
EPOCHS	9
Learning Rate	0.001

We use F1-score, recall, accuracy, and precision as performance measures to assess the efficacy of our algorithms. The computation of these measures is shown in Table 2. True positives (TP) indicate that the prediction is correct and the value is positive; false positives (FP) indicate that the forecast is incorrect but the value is positive; true negatives (TN) indicate that the prediction is true but the value is negative; and false negatives (FN) indicate that the prediction is incorrect and the value is negative in this table.

Table 2. CALCULATION FORMULAS OF PERFORMANCE METRICS

Measure	Formula
Accuracy	$(TP+TN)/(TP+TN+FN+FP)$
Precision	$TP/(TP+FP)$
Recall	$TP/(TP+FN)$
F1-score	$2TP/(2TP+FP+FN)$

Table 3 displays the Accuracy, Precision, Recall, and F1-score values that were acquired from the model-based classifications. We find that across the board, the soft voting ensemble fared better than the two models. Compared to ViT and ResneXt, the soft voting ensemble model achieves an accuracy that is 4% greater. It also had the best F1-Score (97.99%), recall (98%), and accuracy (98.18%) of any model we looked at. Performance measures are compared in Fig. 5.

TABLE 3. CLASSIFICATION RESULTS

Algorithms	Accuracy	Precision	Recall	F1-score
VGG19	0.86	0.8688	0.8600	0.8519
ViT	0.94	0.9436	0.9400	0.9387
DenseNet201	0.86	0.8967	0.8600	0.8600
Inception ResnetV2	0.82	0.8299	0.8200	0.8185
ResneXt	0.88	0.8851	0.8800	0.8752
Hard Voting	0.96	0.9618	0.9600	0.9599
Soft Voting	0.981	0.9818	0.9800	0.9799

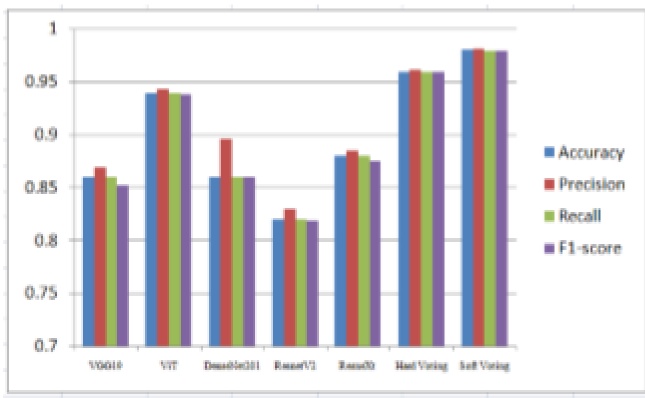

Fig. 5. Comparison of performance metrics

Here, we train on a collection of images that have undergone preprocessing and image enhance techniques. Studying and training the original data allows us to validate the influence of data augmentation on the models. Table 4 displays the experimental outcomes. Table 3 shows that the algorithm's accuracy, precision, recall, and F1 score may be significantly improved by data preparation and data augment approaches. Tables 3 and 4 show that the accuracy of the soft voting classifier went raised from 92% to 98.1% after picture enhancement and preprocessing. Additionally, there has been some improvement to other performance measures.

TABLE 4. CLASSIFICATION RESULTS ON ORIGINAL DATA

Algorithms	Accuracy	Precision	Recall	F1-score
VGG19	0.82	0.8392	0.8200	0.8185
ViT	0.88	0.8857	0.8800	0.8802
DenseNet201	0.86	0.8593	0.8600	0.8595
Inception ResnetV2	0.74	0.7453	0.7400	0.7375
ResneXt	0.88	0.8838	0.8800	0.8802
Hard Voting	0.91	0.9120	0.9100	0.9099
Soft Voting	0.92	0.9219	0.9200	0.9202

Algorithm and classifier performance visualization is a common use case for confusion matrices. The normalized confusion matrix of the model based on the soft voting ensemble is shown in Figure 6. The results demonstrate that it is capable of accurately identifying all test samples for four of the five classes, with an error rate of around 2% for the fifth class.

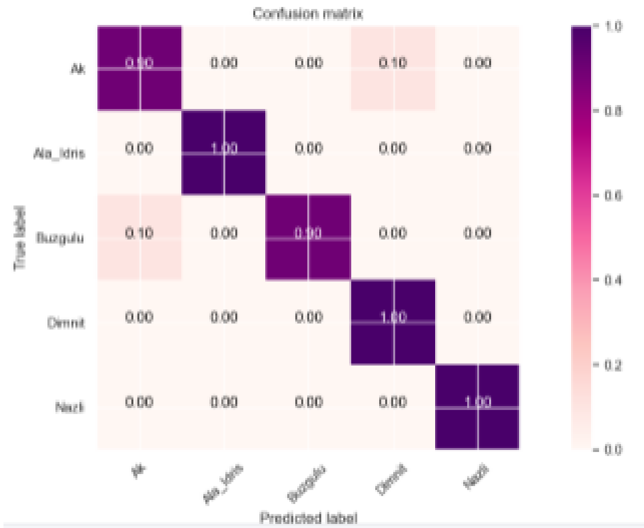

Fig. 6. Normalized confusion matrix of Soft Voting ensemble model

We conclude by comparing the suggested model in [6] and [7] with our own methods in order to assess its merit. The results of the categorization are shown in Table 5. As you can see from the results, the suggested model is superior than the alternatives. Two methods have been proposed for classifying grapevine leaves: one that uses an adaptation of DenseNet201 and another that use a pre-trained MobileNetv2 Logits layer for feature extraction and support vector machine (SVM) kernels for classification.

TABLE 5. CLASSIFICATION RESULTS

`	Accuracy	Precision	Recall	F1-score
M.Koklu[2]	0.976	0.9762	0.760	0.9760
Hunar[1]	0.9802	0.9800	0.9818	0.9800
Hard Voting	0.96	0.9618	0.9600	0.9599
Soft Voting	0.981	0.9818	0.9800	0.9799

IV. CONCLUSION

This work aimed to categorize five species of grapevine leaves using ensembling techniques and transfer learning models like VGG19, VIT, Inception ResnetV2, DenseNet201, ResneXt. Soft voting classifiers using VGG19, VIT, Inception ResnetV2, DenseNet201, and ResneXt outperformed both models discussed in this paper. much if the models perform at the SOTA level, they may be much better. Things to keep in mind are: Since this study's sample size is modest, further data might help enhance identification accuracy and lower the likelihood of misclassification. For the purpose of this work, we used five different deep learning models to identify grape leaves. Actually, EfficientNet and Swin Transformer are only two of several learning models that are capable of accurate object recognition. Using these models to recognize grape leaves is the next logical step in our process. Other ensemble approaches that are used: Here, we classified data using an ensemble approach that included hard voting and soft voting. Actually, a stacked ensemble learning classifier may be trained to recognize grapevine leaves by passing along information learnt from many classification models.

References

[1] K. Simonyan, A. Zisserman. (2014)Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556

[2] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner. (2021)An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In Proceedings of the 2021 International Conference on Learning Representations (ICLP), Colombo, Sri Lanka. pp:20–27.

[3] C. Szegedy, S. Ioffe, V.Vanhoucke, A. Alemi. (2016)Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning, ArXiv, abs/1602.07261

[4] G. Huang, Z. Liu, Van Der Maaten, L. Weinberger, K.Q.(2017)Densely connected convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI. pp:4700-4708.

[5] Xie, R. Girshick, P. Dollár, Z. Tu , K. He.(2017)Aggregated Residual Transformations for Deep Neural Networks, In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI. pp:5987-5995

[6] Ahmed Hunar, Hama Hersh, Jalal, Shayan, Ahmed Mohammed.(2023) Deep Learning in Grapevine Leaves Varieties Classification Based on Dense Convolutional Network. Journal of Image and Graphics, 11:98103.

[7] M. Koklu, M. F. Unlersen, I. A. Ozkan. (2021)A CNN-SVM study based on selected deep features for grapevine leaves classification. Measurement ,188(110425):1-10

[8] TamvakisPetros, KiourtChairi, Solomou Alexandra, Ioannakis George Alexis, Tsirliganis, Nestoras. (2022)Semantic Image Segmentation with Deep Learning for Vine Leaf Phenotyping. IFACPapersOnLine,55: 83-88.

[9] Chen Yiping , Wu Qiufeng.(2022)Grape leaf disease identification with sparse data via generative adversarial networks and convolutional neural networks. Precision Agriculture. 24: 235–253.

[10] Bingpiao Liu, Lufeng Luo, Jinhai Wang, Qinghua Lu, Huiling Wei, Yunzhi Zhang, Wenbo Zhu. (2023)An improved lightweight network based on deep learning for grape recognition in unstructured environments. Information Processing in Agriculture(In Press).

[11] Sozzi, M., Cantalamessa, S., Cogato, A., Kayad, A., F. Marinello. (2022)Automatic Bunch Detection in White Grape Varieties Using YOLOv3, YOLOv4, and YOLOv5 Deep Learning Algorithms. Agronomy,12(2): 319

[12] Saloni Kumari, Deepika Kumar, Mamta Mittal. (2021)An ensemble approach for classification and prediction of diabetes mellitus using soft voting classifier. International Journal of Cognitive Computing in Engineering, 2:40-46