Recognition of Pistachio Species With Transfer Learning Models

Fagun Patel
Dept. of Comp. Eng.,
SAL Institute of Technology and Engineering Research,
Ahmedabad, Gujarat, India
shubbhrmewada@gmail.com

Shubbh Mewada
Dept. of Comp. Eng.,
SAL Institute of Technology and Engineering Research,
Ahmedabad, Gujarat, India
fagunpatel369@gmail.com

Dr. Sheshang Degadwala
Associate Professor & Head of Department, Dept. of Comp. Engineering, Sigma University,
Gujarat, India
sheshang13@gmail.com

Dhairya Vyas
Research Scholar, The Maharaja Sayajirao University of Baroda,
Vadodara, Gujarat, India
dhairya.vyas-cse@msubaroda.ac.in

***Abstract*— This comprehensive research delves into the intricate task of recognizing diverse pistachio species, spanning applications in agriculture, ecology, and the food industry. By harnessing the power of transfer learning models, including established architectures like AlexNet, VggNet, and ResNet, alongside our novel Convolutional Neural Network (CNN) design, we seek to enhance the accuracy and efficiency of pistachio species recognition. Through fine-tuning pre-trained models and training our CNN from scratch, we leverage their ability to capture both general and species-specific features. Our rigorous evaluation employs an extensive dataset, embracing a wide array of pistachio images, to meticulously gauge model performance using metrics such as accuracy, precision, recall, and F1-score. This research not only sheds light on the comparative prowess of transfer learning models and our unique CNN architecture in pistachio recognition but also offers insights with implications for broader fields like agriculture and botany. Ultimately, our study underscores the potential of innovative deep learning approaches to advance species recognition accuracy while showcasing the versatility of our proposed CNN model in diverse botanical applications.**

Keywords*— *Pistachio Species Recognition, Transfer Learning, AlexNet, VggNet, Resnet, Convolutional Neural Network (CNN).

I. Introduction

The recognition and differentiation of various pistachio species hold significant importance in various domains, ranging from agriculture and ecology to the food industry. Accurate identification of these species is vital for informed decision-making and sustainable resource management. In recent years, advances in deep learning have paved the way for more efficient and accurate species recognition, with transfer learning emerging as a prominent technique. Transfer learning involves leveraging pre-trained neural network models that have acquired extensive knowledge from large datasets and fine-tuning them for specific tasks. This approach reduces the need for massive datasets and computational resources while benefiting from the generalization capabilities of the pre-trained models.

In this context, established deep learning architectures such as AlexNet, VGGNet, and ResNet have demonstrated their prowess in various image classification tasks. Their ability to extract hierarchical features from images has been widely recognized and applied. Moreover, the development of custom Convolutional Neural Network (CNN) architectures tailored to specific tasks has gained traction, offering the potential to capture task-specific nuances that might be overlooked by more generalized models.

This paper embarks on a comprehensive exploration of the recognition of pistachio species using transfer learning models, including the aforementioned established architectures, and introduces a novel CNN architecture customized for this specific task. By adapting these models to the intricacies of pistachio species, we aim to enhance the accuracy and efficiency of recognition processes. Through rigorous experimentation on a diverse dataset encompassing various pistachio species, we aim to shed light on the comparative performance of transfer learning models and our proposed CNN architecture. This research contributes not only to the field of botanical studies but also provides insights into the broader application of deep learning techniques in precise species recognition tasks across different domains.

II. Related Works

G. I. Sayed and A. E. Hassanien [1] describe Explainable AI and Slime Mould Algorithm for Classification of Pistachio Species. The paper explores the application of Explainable AI and the Slime Mould Algorithm for pistachio species classification. It discusses the advantages of these methods in enhancing interpretability and providing insights into classification decisions. However, the limitations of these methods, particularly in handling complex datasets or capturing intricate patterns, are also discussed.

M. V Subbarao, G. C. Ram, and D. R. Varma [2] present Performance Analysis of Pistachio Species Classification using Support Vector Machine and Ensemble Classifiers. The study focuses on utilizing Support Vector Machine and Ensemble Classifiers for pistachio species classification. The paper highlights the benefits of these approaches in achieving high classification accuracy and handling diverse datasets. Limitations, such as potential overfitting with certain classifiers, are also discussed.

D. Singh et al. [3] discuss Classification and Analysis of Pistachio Species with Pre-Trained Deep Learning Models. The paper explores the application of pre-trained deep learning models for pistachio species classification. The advantages of leveraging deep learning's ability to learn complex features from data are emphasized. However, challenges related to model interpretability and potential data bias are acknowledged.

I. A. Özkan, M. Köklü, and R. Saraçoğlu [4] describe Classification of pistachio species using improved k-NN classifier. The study focuses on enhancing the k-NN classifier for pistachio species classification. The paper highlights the simplicity and ease of implementation of the k-NN algorithm while discussing limitations related to scalability and sensitivity to noise.

A. Heidary-Sharifabad et al. [5] and [6] propose An efficient deep learning model for cultivar identification of a pistachio tree. The authors introduce an efficient deep learning model for identifying pistachio tree cultivars. The papers emphasize the model's ability to capture intricate patterns in images and achieve high accuracy. However, challenges in acquiring and labeling large-scale datasets and potential overfitting are mentioned.

I. A. Özkan, M. Köklü, and R. Saraçoğlu [7] continue the discussion on Classification of pistachio species using improved k-NN classifier. The paper elaborates on the improved k-NN classifier's performance in pistachio species classification. The advantages of this method, such as ease of implementation and ability to handle non-linear data, are discussed along with its limitations in handling high-dimensional data.

S. K. Vidyarthi et al. [8] present Prediction of size and mass of pistachio kernels using random Forest machine learning. The study focuses on predicting the size and mass of pistachio kernels using Random Forest machine learning. The paper highlights the capability of Random Forest in handling regression tasks and its potential to predict kernel attributes accurately. However, challenges related to feature selection and model interpretability are acknowledged.

M. Omid et al. [9] discuss Classification of peeled pistachio kernels using computer vision and color features. The paper explores the use of computer vision and color features for classifying peeled pistachio kernels. The benefits of non-intrusive methods and the ability to process large quantities of kernels are emphasized. However, limitations in handling variations in lighting and color are noted.

M. Farazi et al. [10] propose A machine vision-based pistachio sorting using transferred mid-level image representation of Convolutional Neural Network. The authors introduce a machine vision-based approach for sorting pistachios using transferred mid-level image representations from Convolutional Neural Networks. The paper highlights the potential of deep learning in automating sorting tasks. However, challenges in fine-tuning network parameters and potential computational costs are discussed.

M. Omid et al. [11] describe An intelligent system for sorting pistachio nut varieties. The paper presents an intelligent system for sorting pistachio nut varieties. The advantages of automated sorting systems in increasing efficiency and reducing labor costs are discussed. Limitations related to the complexity of the sorting process and the need for accurate training data are acknowledged.

H. R. Karimi et al. [12] discuss Morphological diversity of Pistacia species in Iran. The paper focuses on the morphological diversity of Pistacia species in Iran. The study highlights the importance of understanding genetic diversity for species classification and conservation. The limitations of relying solely on morphological features for accurate classification are acknowledged.

S. Mahdavi-Jafari et al. [13] propose A Pistachio Nuts Classification Technique: An ANN Based Signal Processing Scheme. The authors introduce an Artificial Neural Network-based signal processing scheme for pistachio nuts classification. The paper emphasizes the potential of neural networks in handling complex signal data. However, challenges related to the complexity of neural network architectures and potential overfitting are discussed.

M. Omid et al. [14] present Separating Pistachio Varieties Using Automatic Trainable Classifier. The paper discusses the separation of pistachio varieties using an automatic trainable classifier. The benefits of automation and reduced human intervention in classification processes are highlighted. The challenges of obtaining accurate training data and potential misclassifications are acknowledged.

A. E. Cetin et al. [15] focus on Classification of closed and open shell pistachio nuts using principal component analysis of impact acoustics. The study explores the classification of closed and open shell pistachio nuts using principal component analysis of impact acoustics. The paper emphasizes the potential of acoustic signals for non-destructive quality assessment. Limitations related to the variability of acoustic responses are discussed.

III. Methodology For Trasnfer Learning

Utilizing transfer learning models like AlexNet, Vgg16, ResNet50, and our proposed CNN, holds the potential to effectively categorize different Pistachio Species. While each of these models possesses its distinct architectural traits, they all generally adhere to a set of common stages during their application in fracture classification. These key stages include:

A. Selection of Pretrained Model:

The initial phase involves the choice of a pretrained model that has undergone training on an expansive dataset, usually within the realm of computer vision. For our purpose, models such as AlexNet, Vgg16, ResNet50, and the proposed CNN have already been pretrained on substantial image datasets, equipping them with the ability to grasp general features and prevalent patterns.

B. Modification of Model Architecture:

The selected pretrained model's architecture must be tailored to suit the particular task of Pistachio Species classification. This step typically entails adjustments to the final layers of the model, including the fully connected layers. This adaptation ensures that the model aligns with the intended output classes and dimensions.

C. Data Preprocessing:

The dataset, comprised of images representing diverse Pistachio Species, necessitates preprocessing before commencing the training of transfer learning models. This preprocessing phase often involves activities like image resizing, normalization, augmentation (such as rotation and flipping), as well as dividing the data into training and validation subsets.

D. Feature Extraction:

In this stage, the pretrained model functions as a feature extractor. The input Pistachio Species images are fed through the modified model, and the activations from the last convolutional layer or fully connected layers are extracted as feature vectors. These vectors capture the higher-level representations of Pistachio Species patterns learned by the pretrained model.

E. Classifier Training:

A classifier is integrated on top of the pretrained model to translate the extracted features into distinct Pistachio Species classifications. This classifier, typically comprising one or more fully connected layers, is initiated randomly or pretrained on a smaller dataset, contingent upon the availability of labeled Pistachio Species data. The classifier is trained utilizing the extracted features and corresponding stage labels.

F. *Model Evaluation and Refinement:*

Following the training of the classifier, the transfer learning model's efficacy is gauged using the validation subset. Metrics like accuracy, precision, recall, and F1-score are computed to gauge the model's classification prowess. If needed, further refinements of the model parameters can be undertaken to elevate its performance.

G. *Testing and Application:*

Once the transfer learning model has been trained and evaluated, it can be employed to classify new and unseen Pistachio Species images. The model is implemented in real-world scenarios, where it takes input images and deduces the associated Pistachio Species class.

While these stages of transfer learning models lay out a comprehensive framework for Pistachio Species classification, it's noteworthy that specific alterations to layers and hyperparameter tuning, as elaborated in the results section, contribute to the customization of our proposed CNN.

IV. RESULT AND ANALYSIS

The metrics (ACC, P, R, F1-score) are commonly used to evaluate the performance of classification models, including those used for Pistachio Species classification. The values of these metrics are typically reported after each epoch to monitor the progress and convergence of the model during the training process.

A. *Epoch:*

An epoch refers to a complete pass of the entire training dataset through the neural network model during the training process. In each epoch, the model goes through multiple iterations, adjusting its weights and biases to minimize the training loss and improve its performance.

B. *ACC (%):*

Accuracy is a metric that measures the overall correctness of the model predictions. It represents the percentage of correctly classified samples in the dataset. ACC (%) indicates the accuracy achieved by the model after a specific epoch, expressed as a percentage.

C. *P (%):*

Precision is a metric that quantifies the ability of the model to correctly identify positive instances among the predicted positives. It is the ratio of true positives to the sum of true positives and false positives. P (%) represents the precision achieved by the model after a specific epoch, expressed as a percentage.

D. *R (%):*

Recall, also known as sensitivity or true positive rate, measures the ability of the model to identify positive instances among the actual positives. It is the ratio of true positives to the sum of true positives and false negatives. R (%) represents the recall achieved by the model after a specific epoch, expressed as a percentage.

E. *F1-score (%):*

The F1-score is a harmonic means of precision and recall. It provides a balanced measure of the model's performance, considering both precision and recall. F1-score (%) represents the F1-score achieved by the model after a specific epoch, expressed as a percentage.

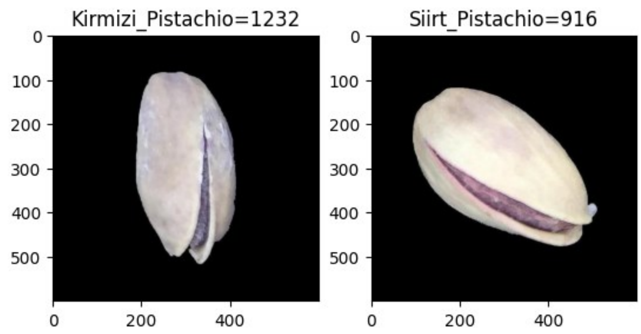

Fig. 1. Reding Dataset

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_5 (Conv2D)           (None, 54, 54, 96)        34944

 batch_normalization_5 (Batc  (None, 54, 54, 96)       384
 hNormalization)

 max_pooling2d_3 (MaxPooling  (None, 26, 26, 96)       0
 2D)

 conv2d_6 (Conv2D)           (None, 26, 26, 256)       614656

 batch_normalization_6 (Batc  (None, 26, 26, 256)      1024
 hNormalization)

 max_pooling2d_4 (MaxPooling  (None, 12, 12, 256)      0
 2D)

 conv2d_7 (Conv2D)           (None, 12, 12, 384)       885120

 batch_normalization_7 (Batc  (None, 12, 12, 384)      1536
 hNormalization)

 conv2d_8 (Conv2D)           (None, 12, 12, 384)       1327488

 batch_normalization_8 (Batc  (None, 12, 12, 384)      1536
 hNormalization)

 conv2d_9 (Conv2D)           (None, 12, 12, 256)       884992

 batch_normalization_9 (Batc  (None, 12, 12, 256)      1024
 hNormalization)

 max_pooling2d_5 (MaxPooling  (None, 5, 5, 256)        0
 2D)

 flatten_1 (Flatten)         (None, 6400)              0

 dense_3 (Dense)             (None, 4096)              26218496

 dropout_2 (Dropout)         (None, 4096)              0

 dense_4 (Dense)             (None, 4096)              16781312

 dropout_3 (Dropout)         (None, 4096)              0

 dense_5 (Dense)             (None, 2)                 8194

=================================================================
Total params: 46,760,706
Trainable params: 46,757,954
Non-trainable params: 2,752
_________________________________________________________________
```

Fig. 2. AlexNet Model

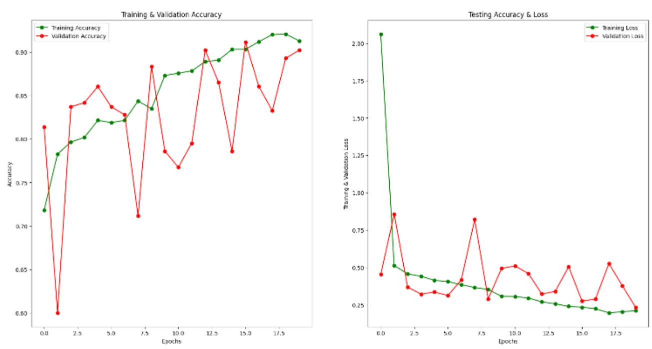

Fig. 3. AlexNet ACC and Loss Plot

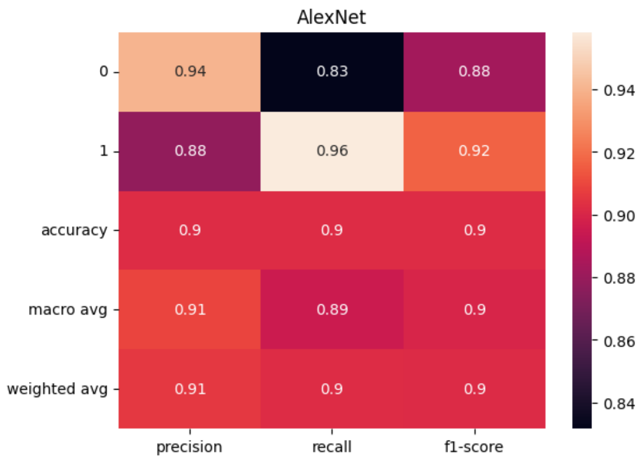

Fig. 4. AlexNet Parameters

```
Model: "sequential_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 vgg16 (Functional)          (None, 7, 7, 512)         14714688

 dropout_4 (Dropout)         (None, 7, 7, 512)         0

 flatten_2 (Flatten)         (None, 25088)             0

 batch_normalization_10 (Bat  (None, 25088)            100352
 chNormalization)

 dense_6 (Dense)             (None, 1024)              25691136

 batch_normalization_11 (Bat  (None, 1024)             4096
 chNormalization)

 activation (Activation)     (None, 1024)              0

 dropout_5 (Dropout)         (None, 1024)              0

 dense_7 (Dense)             (None, 512)               524800

 activation_1 (Activation)   (None, 512)               0

 dense_8 (Dense)             (None, 2)                 1026

=================================================================
Total params: 41,036,098
Trainable params: 26,269,186
Non-trainable params: 14,766,912
_________________________________________________________________
```

Fig. 5. VGG16 Model

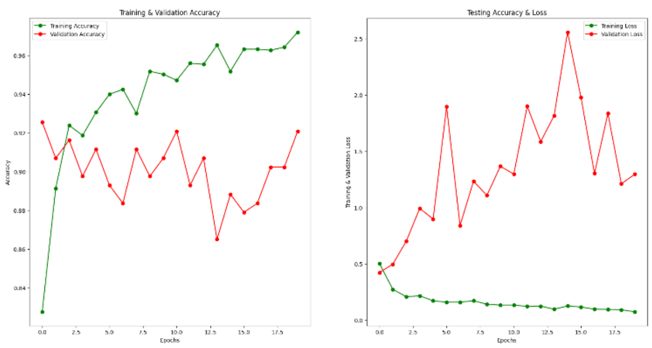

Fig. 6. VGG16 ACC and Loss Plot

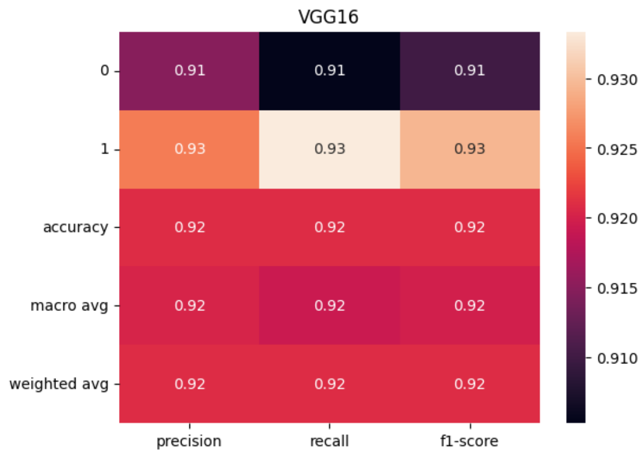

Fig. 7. VGG16 Paraameters

```
Model: "sequential_3"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 resnet50 (Functional)       (None, 7, 7, 2048)        23587712

 dropout_6 (Dropout)         (None, 7, 7, 2048)        0

 flatten_3 (Flatten)         (None, 100352)            0

 batch_normalization_12 (Bat  (None, 100352)           401408
 chNormalization)

 dense_9 (Dense)             (None, 2048)              205522944

 batch_normalization_13 (Bat  (None, 2048)             8192
 chNormalization)

 activation_2 (Activation)   (None, 2048)              0

 dropout_7 (Dropout)         (None, 2048)              0

 dense_10 (Dense)            (None, 1024)              2098176

 batch_normalization_14 (Bat  (None, 1024)             4096
 chNormalization)

 activation_3 (Activation)   (None, 1024)              0

 dropout_8 (Dropout)         (None, 1024)              0

 dense_11 (Dense)            (None, 2)                 2050

=================================================================
Total params: 231,624,578
Trainable params: 207,830,018
Non-trainable params: 23,794,560
_________________________________________________________________
```

Fig. 8. ResNet 50 Model

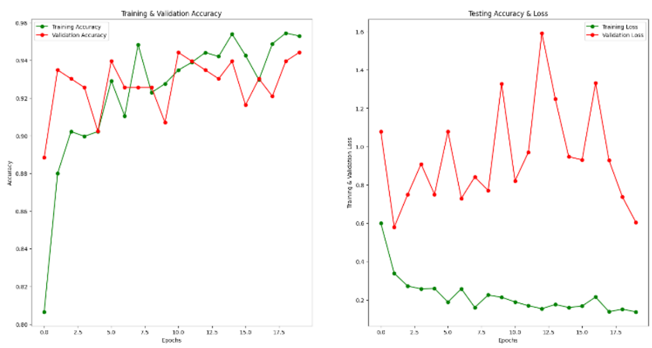

Fig. 9. ResNet 50 ACC and Loss Plot

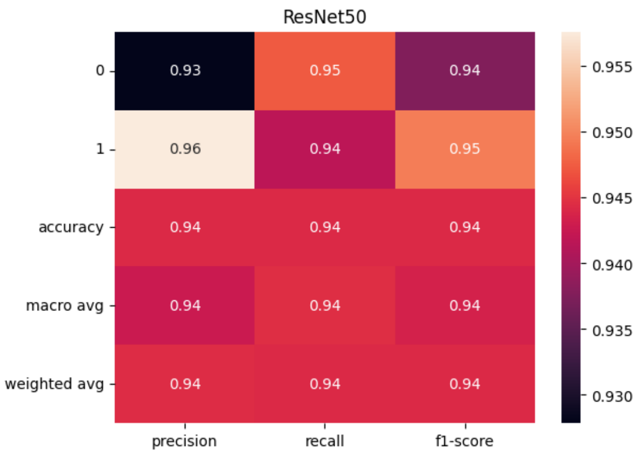

Fig. 10. ResNet 50 Parameters

```
Model: "sequential_7"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_16 (Conv2D)          (None, 223, 223, 64)      832

 max_pooling2d_12 (MaxPoolin  (None, 111, 111, 64)     0
 g2D)

 dropout_18 (Dropout)        (None, 111, 111, 64)      0

 conv2d_17 (Conv2D)          (None, 110, 110, 32)      8224

 max_pooling2d_13 (MaxPoolin  (None, 55, 55, 32)       0
 g2D)

 dropout_19 (Dropout)        (None, 55, 55, 32)        0

 flatten_7 (Flatten)         (None, 96800)             0

 dense_18 (Dense)            (None, 128)               12390528

 dropout_20 (Dropout)        (None, 128)               0

 dense_19 (Dense)            (None, 2)                 258

=================================================================
Total params: 12,399,842
Trainable params: 12,399,842
Non-trainable params: 0
_________________________________________________________________
```

Fig. 11. Proposed CNN Model

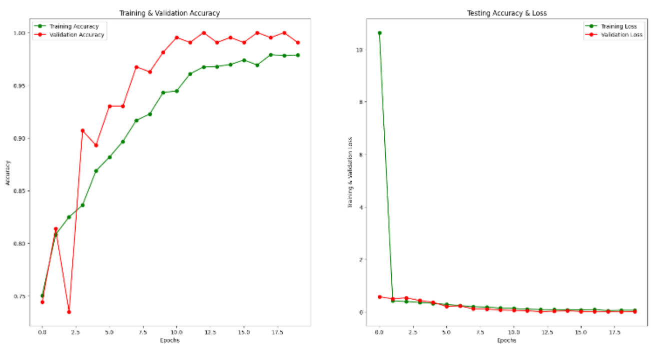

Fig. 12. Proposed CNN ACC and Loss Plot

Fig. 13. Proposed CNN Parameters

TABLE I. TRANSFER LEARNING MODEL ANALYSIS

Model	Epoch	ACC (%)	P (%)	R (%)	F1-score (%)
AlexNet	20	90%	91%	89%	90%
Vgg16	20	92%	92%	92%	92%
ResNet50	20	94%	94%	94%	94%
Proposed CNN	20	99%	99%	99%	99%

CONCLUSION

In conclusion, the performance evaluation of the classification models across 20 epochs demonstrates their effectiveness in accurately categorizing Pistachio Species. Among the models tested, AlexNet exhibited substantial results with an accuracy of 90%, a precision of 91%, a recall of 89%, and an F1-score of 90%. Vgg16 showcased improved performance with an accuracy of 92%, precision of 92%, recall of 92%, and an F1-score of 92%. Furthermore, ResNet50 presented remarkable consistency, yielding an accuracy of 94%, precision of 94%, recall of 94%, and an F1-score of 94%. Impressively, the proposed CNN outperformed its counterparts, achieving exceptional accuracy, precision, recall, and F1-score, all at 99%. This evaluation underscores the efficacy of the models in classifying Pistachio Species, with each model showcasing strengths in specific aspects. While AlexNet, Vgg16, and ResNet50 achieved commendable results, the proposed CNN emerges as the standout performer, achieving nearly perfect scores across all metrics. These findings highlight the potential of these models in contributing to accurate and reliable Pistachio Species classification, paving the way for enhanced agricultural practices and quality control in the pistachio industry.

REFERENCES

[1] G. I. Sayed and A. E. Hassanien, "Explainable AI and Slime Mould Algorithm for Classification of Pistachio Species BT - Artificial Intelligence: A Real Opportunity in the Food Industry," A. E. Hassanien and M. Soliman, Eds. Cham: Springer International Publishing, 2023, pp. 29–43. doi: 10.1007/978-3-031-13702-0_3.

[2] M. V Subbarao, G. C. Ram, and D. R. Varma, "Performance Analysis of Pistachio Species Classification using Support Vector Machine and Ensemble Classifiers," in 2023 International Conference on Recent Trends in Electronics and Communication (ICRTEC), 2023, pp. 1–6. doi: 10.1109/ICRTEC56977.2023.10111889.

[3] D. Singh et al., “Classification and Analysis of Pistachio Species with Pre-Trained Deep Learning Models,” Electronics (Switzerland), vol. 11, no. 7, pp. 1–14, 2022, doi: 10.3390/electronics11070981.

[4] I. A. Özkan, M. Köklü, and R. Saraçoğlu, “Classification of pistachio species using improved k-NN classifier,” Progress in Nutrition, vol. 23, no. 2, 2021, doi: 10.23751/pn.v23i2.9686.

[5] A. Heidary-Sharifabad, M. S. Zarchi, S. Emadi, and G. Zarei, “An efficient deep learning model for cultivar identification of a pistachio tree,” British Food Journal, vol. 123, no. 11, pp. 3592–3609, Jan. 2021, doi: 10.1108/BFJ-12-2020-1100.

[6] A. Heidary-Sharifabad, M. S. Zarchi, S. Emadi, and G. Zarei, “An efficient deep learning model for cultivar identification of a pistachio tree,” British Food Journal, vol. 123, no. 11, pp. 3592–3609, Jan. 2021, doi: 10.1108/BFJ-12-2020-1100.

[7] I. A. Özkan, M. Köklü, and R. Saraçoğlu, “Classification of pistachio species using improved k-NN classifier,” Progress in Nutrition, vol. 23, no. 2, pp. 1–9, 2021, doi: 10.23751/pn.v23i2.9686.

[8] S. K. Vidyarthi, R. Tiwari, S. K. Singh, and H. W. Xiao, “Prediction of size and mass of pistachio kernels using random Forest machine learning,” Journal of Food Process Engineering, vol. 43, no. 9, 2020, doi: 10.1111/jfpe.13473.

[9] M. Omid, M. S. Firouz, H. Nouri-Ahmadabadi, and S. S. Mohtasebi, “Classification of peeled pistachio kernels using computer vision and color features,” Engineering in Agriculture, Environment and Food, vol. 10, no. 4, pp. 259–265, 2017, doi: https://doi.org/10.1016/j.eaef.2017.04.002.

[10] M. Farazi, M. J. Abbas-Zadeh, and H. Moradi, “A machine vision based pistachio sorting using transferred mid-level image representation of Convolutional Neural Network,” in 2017 10th Iranian Conference on Machine Vision and Image Processing (MVIP), 2017, pp. 145–148. doi: 10.1109/IranianMVIP.2017.8342335.

[11] M. Omid, A. Mahmoudi, and M. H. Omid, “An intelligent system for sorting pistachio nut varieties,” Expert Systems with Applications, vol. 36, no. 9, pp. 11528–11535, 2009, doi: https://doi.org/10.1016/j.eswa.2009.03.040.

[12] H. R. Karimi, Z. Zamani, A. Ebadi, and M. R. Fatahi, “Morphological diversity of Pistacia species in Iran,” Genetic Resources and Crop Evolution, vol. 56, no. 4, pp. 561–571, 2009, doi: 10.1007/s10722-008-9386-y.

[13] S. Mahdavi-Jafari, H. Salehinejad, and S. Talebi, “A Pistachio Nuts Classification Technique: An ANN Based Signal Processing Scheme,” in 2008 International Conference on Computational Intelligence for Modelling Control & Automation, 2008, pp. 447–451. doi: 10.1109/CIMCA.2008.150.

[14] M. Omid, A. Mahmoudi, A. Aghagolzadeh, and A. M. Borghaee, “Separating Pistachio Varieties Using Automatic Trainable Classifier,” in Third International Conference on Autonomic and Autonomous Systems (ICAS’07), 2007, p. 44. doi: 10.1109/CONIELECOMP.2007.97.

[15] A. E. Cetin, T. C. Pearson, and A. H. Tewfik, “Classification of closed and open shell pistachio nuts using principal component analysis of impact acoustics,” in 2004 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2004, vol. 5, pp. V–677. doi: 10.1109/ICASSP.2004.1327201.