iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 12 **Issue:** VI **Month of publication:** June 2024

DOI: https://doi.org/10.22214/ijraset.2024.63483

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue VI June 2024- Available at www.ijraset.com

A Hybrid Classification Model (Fruits or Vegetable) Using Deep Learning Techniques

Karan Kumar Maurya[1], Adarsh Verma[2], Danish Gaur[3], Ankit Patel[4]
Quantum School of Technology, Quantum University, Roorkee, Uttarakhand 247167 Department of Computer Science and Engineering

Abstract: ***In modern vision and pattern recognition, complex tasks such as picture analysis, facial recognition, fingerprint identification, and DNA sequencing necessitate a nuanced approach, often requiring the integration of multiple feature descriptors. This research proposes a multi-model identification and classification strategy leveraging multi- feature fusion techniques to address these intricate challenges. Specifically, the focus is on fruit and vegetable recognition and classification, a burgeoning field in computer and machine vision. By employing an identification system tailored to fruits and vegetables and harnessing the capabilities of MobileNetV2 architecture, customers and buyers can more easily discern the type and quality of produce. MobileNetV2, a convolutional neural network architecture optimized for mobile devices, offers promising performance in real-world applications. This abstract highlight the significance of CNNs and MobileNetV2 in tackling multifaceted recognition tasks, underscoring the potential for enhanced efficiency and accuracy in fruit and vegetable classification.***
Keywords: CNNs, Convolutional Neural Networks, MobileNetV2, Fruit and Vegetable Recognition, Multi-Feature Fusion.

I. INTRODUCTION

In the pursuit of a healthy lifestyle, the inclusion of fruits and vegetables is paramount due to their myriad health benefits. However, the accessibility of certain produce is often dictated by seasonal variations, presenting a challenge for consumers and marketers alike. In India, where agriculture remains a cornerstone of the national economy, a staggering seventy percent of land is dedicated to cultivation. Remarkably, India boasts the third position globally in fruit production and the second in vegetable production.

In this landscape, the integration of deep learning algorithms for fruit and vegetable categorization emerges as a significant boon for both marketers and consumers. The burgeoning reliance on computer science and information technology within the agricultural sector further amplifies the relevance of such techniques. With the advent of artificial intelligence and soft computing-based methodologies, the provision of high-quality produce to consumers has become increasingly streamlined.

Convolutional Neural Networks (CNNs) have revolutionized image classification tasks due to their ability to automatically learn features from raw data. Here's an overview of some key CNN techniques and their applications in image classification:

1) Convolutional Layers: These layers apply filters or kernels to the input image, performing convolutions to extract features. Convolutional operations capture spatial hierarchies of patterns, allowing the network to learn features at different levels of abstraction.
2) Pooling Layers: Pooling layers reduce the spatial dimensions of the feature maps generated by the convolutional layers. Max pooling and average pooling are commonly used techniques to down sample the feature maps while retaining important information.
3) Activation Functions: Activation functions like ReLU (Rectified Linear Unit), Leaky ReLU, or variants like ELU (Exponential Linear Unit) introduce non-linearity into the network, enabling it to learn complex relationships in the data.
4) Normalization Layers: Techniques like Batch Normalization normalize the activations of each layer, improving the stability and convergence speed of the network during training.
5) Data Augmentation: Data augmentation techniques such as rotation, translation, scaling, and flipping are applied to increase the diversity of the training dataset. This helps prevent overfitting and improves the generalization ability of the model.
6) Transfer Learning: Transfer learning involves leveraging pre-trained CNN models trained on large datasets like ImageNet. Fine-tuning these models on specific image classification tasks with smaller datasets can significantly boost performance and reduce training time.

II. LITERATURE REVIEW

Generally, the ML approach is extracting the image feature-metrics for the classification. This includes various processing steps such as image preprocessing, feature extraction, classification model development, and validation (Jana et.al.,2017; Hou et.al.,2016. Azizah,2017; Thenmozhi et.al.,2019). Much work related to fruit classification has been put forward. Support Vector Machines (SVM) display acceptable results on small data sets. This can be done flawlessly using deep learning based defined neural networks (Saranya et.al., 2020).

CNN based model improves image classification for large datasets. The CNN based model (Palakodati et.al., 2020) has achieved accuracy of 97.82% in classifying fresh and rotten category of fruits. Three Convolution layers, three Max pooling, one fully connected layer, and a SoftMax classifier were used to achieve the mentioned accuracy in 225 epochs. In comparison to a few transfer learning models, this model performed better.

The fruit recognition rate was evaluated using only CNN and CNN with a selective algorithm (Hou et.al.,2016). CNN with optional algorithm is proven better than when used CNN alone. Although efficient recognition rate is achieved but work has been done on small classes of fruits and changes in the external environment and other factors such as light are not considered in creation of database.

The CNN model (Azizah et.al.,2017) was used for detecting the defect on mangosteen with accuracy of 97%. Before applying CNN for classification experts manually perform sorting of mangosteen. Image classification by CNN includes 4-fold cross validation process.

Fungus detection and discrimination between various kinds of fungus is done by CNN architecture with 11 layers (Tahir et.al.,2018). CNN layers include 3 convolution, 3 ReLU, 3 pooling and 2 fully connected layers. Accuracy of 94.8% is achieved with 5-fold validation. Fine tuning between different parameters has been done for better results.

Crop insect classification is a big challenge (Thenmozhi et al., 2019), and to solve it, deep CNN models were employed on NBAIR, Xie1 and Xie2 datasets. With the NBAIR, Xie1 and Xie2 datasets, accuracy in insect categorization was 96.75 percent,
97.47 percent, and 95.97 percent, respectively. With the same datasets, several transfer learning models (AlexNet, ResNet-50, ResNet-101, VGG-16, and VGG-19) were employed in insect classification. When compared to the transfer learning model, the CNN model was shown to be more efficient in this study.

CNN is frequently employed in agriculture for picture categorization of a variety of issues (Kamilaris et.al.,2018). The current study shows that a CNN model based on deep learning is more effective in classifying fruits as "fresh" or "rotten." The suggested model's accuracy is also compared to that of other transfer learning methods. Six classes were created from three different varieties of fruit. That is, each fruit is classified as either fresh or rotten. The VGG16, VGG19, AlexNet, and LeNet-5 transfer learning models are investigated. When compared to existing defined models, our system displays a robust CNN model that has enhanced the accuracy for fresh and rotten fruit classification. For better outcomes, we additionally look at the impact of other hyper parameters.

III. METHODOLOGY

Dataset The current work uses dataset “Fruits and Vegetables Image Recognition” in fruit and vegetables classification process. This data set is acquired from Kaggle and has been engineered by collecting, separating, and then labelled. This dataset have 36 classes, almost 100 images for each class so we can say we have 3600+ training images. We have 10 images for each category in train/validation.

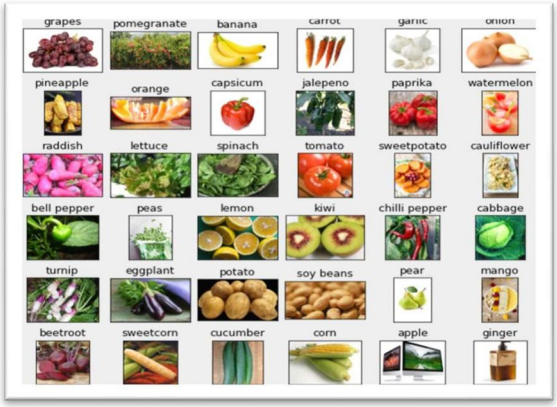


Figure 1:Example of Images in Dataset (Fruits and Vegetables)

A. *Convolutional Neural Network*

Convolution neural networks (CNN) are today's most popular class of models for image recognition and classification. One of the big advantages of using CNN is that it requires much less preprocessing time as compared with other classification algorithm. To improve the classification process, it processes the input data, gives training to model and then takeout the important information automatically. The primary purpose of a CNN algorithm is to download data in a managed format without losing important features in understanding what the data represents. This makes it suitable for working with large data sets. CNN is composed of mainly three layers. The number of layers varies depending on complexity of the problem domain. In complex applications, the number of such layers increases significantly. The image goes through these series of layers, first is convolutional layer, next is pooling layer and finally fully connected layer. After that it generates the output.

In convolution layer filters are applied to the original image. It extracts features from the image. Most of the user-specified parameters i.e. numbers of kernels and size of the kernel are found in the convolution layer. Max pooling or average pooling is performed via pooling layers. In most pooling layers, the maximum pool technique is employed. They're often employed to shrink the size of a network. The completely linked layers are the network's final layers. The output from the previous pool or convolution layer is used as the input for this layer. It classifies the input image into distinct labelled classes using the SoftMax activation function. Figure 2 represents Basic composition of CNN.

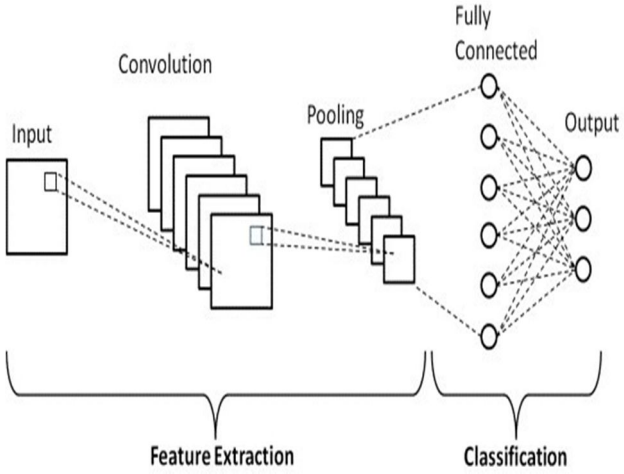


Figure 2:Basic CNN Structure for Classification

B. *MobileNetV2*

In the subsequent study, we employed CNN Model MobileNetV2. Convolutional neural network (CNN) architecture MobileNet-v2 is intended for mobile image categorization applications. It is an excellent option for devices with constrained resources since it is a lightweight model with an effective design that can maintain high accuracy. There are two different kinds of blocks in MobileNetV2. One has a stride of one and is a residual block. Another is a shrinking block with a stride of two.

Each of the two types of blocks has three layers:

1) The first layer uses ReLU6 for 1×1 convolution.
2) The second layer uses depthwise convolution.
3) Another 1×1 convolution, this one without any non-linearity, makes up the third layer. It is asserted that deep networks only have the ability of a linear classifier on the non-zero volume portion of the output domain if ReLU is applied once more.
4) Furthermore, an expansion factor t exists. t=6 for each of the primary experiments.
5) The internal output would have 64×t=64×6=384 channels if the input had 64 channels.

where t: expansion factor, c: number of output channels, n: repeating number, s: stride. 3×3 kernels are used for spatial convolution.

In typical, the primary network (width multiplier 1, 224×224), has a computational cost of 300 million multiply-adds and uses 3.4 million parameters. (Width multiplier is introduced in MobileNetV1.)

The performance trade offs are further explored, for input resolutions from 96 to 224, and width multipliers of 0.35 to 1.4.

The network computational cost up to 585M MAdds, while the model size vary between 1.7M and 6.9M parameters.

To train the network, 16 GPU is used with batch size of 96.

Table 1:Overall Architecture of MobileNetV2

Input	Operator	t	c	n	s
$224^2 \times 3$	Convo2d	-	32	1	2
$112^2 \times 32$	bottleneck	1	16	1	1
$112^2 \times 16$	bottleneck	6	24	2	2
$56^2 \times 24$	bottleneck	6	32	3	2
$28^2 \times 32$	bottleneck	6	64	4	2
$14^2 \times 64$	bottleneck	6	96	3	1
$14^2 \times 96$	bottleneck	6	160	3	2
$7^2 \times 160$	bottleneck	6	320	1	1
$7^2 \times 320$	convo2d 1x1	-	1280	1	1
$7^2 \times 1280$	avgpool 7x7	-	-	1	-
$1 \times 1 \times 1280$	convo2d 1x1	-	k	-	-

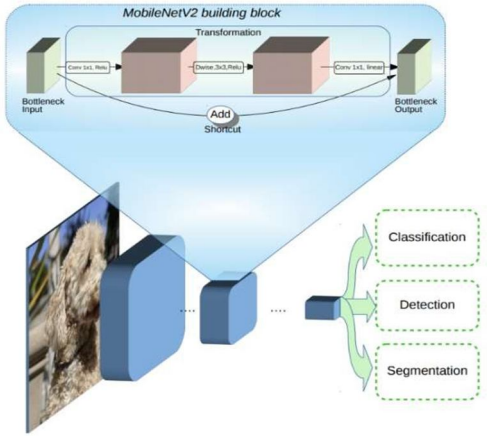


Figure 3:Overview of MobileNetV2

C. Tools and Libraries Keras

Keras is a high-level, deep learning API developed by Google for implementing neural networks. It is written in Python and is used to make the implementation of neural networks easy. It also supports multiple backend neural network computation. We are using Keras for creating model, predicting the object, etc.

1) Numpy

NumPy is a powerful Python library for scientific computing. It provides N- dimensional arrays, mathematical functions, linear algebra tools, random number generation, and efficient element-wise operations. With NumPy, you can handle large datasets, perform complex calculations, and optimize code speed. It's a fundamental tool for researchers, developers, and data scientists. We are using Numpy for the image matrix handling.

2) Streamlit

Streamlit is an open-source Python framework that transforms data scripts into web apps effortlessly. With just a few lines of code, you can create interactive data applications, display model outputs, visualize data, and modify inputs—all without needing front-end experience. It is use to create the web application for the project.

3) BeautifulSoup

Beautiful Soup is a Python library that simplifies web scraping by extracting information from HTML and XML files. It sits on top of an HTML or XML parser, providing Pythonic ways to navigate, search, and modify the parse tree. Whether you're pulling data from websites or processing XML data, Beautiful Soup saves programmers hours of work. We are using it to extract nutritional value of the predicted object from webpage.

4) Requests

Requests is an elegant and simple Python library for making HTTP requests. It provides a straightforward API for interacting with web servers, allowing you to perform operations like GET and POST effortlessly.

5) Pillow

Pillow is a Python library that adds image processing capabilities to your interpreter. It supports many file formats, has a fast internal representation, and offers powerful image operations.

IV. RESULTS

The present work includes classification of Fruits and Vegetables. Dataset utilized in this study is "Fruit and Vegetable Image Recognition". Firstly, the dataset is partitioned into three categories. 80% dataset has used in training while the remaining 10% in validation and 10% used for testing. Validation and training of the same is done simultaneously. In the training process impact of various hyper parameters (mentioned in discussion section) were analysed and adjusted to get a precise model as compared with pre-trained transfer learning models. We used a Lenovo Legion 5 with a Ryzen 5 4000 series, 8 GB RAM, 1 TB HDD + 256 SDD, and NVIDIA GTX 1650 graphic card to run this deep CNN-based model.

Outputs

```
Epoch 1/9
87/87 ━━━━━━━━━━━━━━━━━━━━ 134s 1s/step - accuracy: 0.3365 - loss: 2.6008 - val_accuracy: 0.8533 - val_loss: 0.4916
Epoch 2/9
87/87 ━━━━━━━━━━━━━━━━━━━━ 123s 1s/step - accuracy: 0.8153 - loss: 0.6046 - val_accuracy: 0.9281 - val_loss: 0.2954
Epoch 3/9
87/87 ━━━━━━━━━━━━━━━━━━━━ 123s 1s/step - accuracy: 0.8995 - loss: 0.3373 - val_accuracy: 0.9461 - val_loss: 0.2373
Epoch 4/9
87/87 ━━━━━━━━━━━━━━━━━━━━ 123s 1s/step - accuracy: 0.9403 - loss: 0.1973 - val_accuracy: 0.9251 - val_loss: 0.2264
Epoch 5/9
87/87 ━━━━━━━━━━━━━━━━━━━━ 124s 1s/step - accuracy: 0.9514 - loss: 0.1536 - val_accuracy: 0.9551 - val_loss: 0.1669
Epoch 6/9
87/87 ━━━━━━━━━━━━━━━━━━━━ 124s 1s/step - accuracy: 0.9686 - loss: 0.0983 - val_accuracy: 0.9581 - val_loss: 0.1643
Epoch 7/9
87/87 ━━━━━━━━━━━━━━━━━━━━ 123s 1s/step - accuracy: 0.9769 - loss: 0.0855 - val_accuracy: 0.9671 - val_loss: 0.1349
Epoch 8/9
87/87 ━━━━━━━━━━━━━━━━━━━━ 123s 1s/step - accuracy: 0.9854 - loss: 0.0605 - val_accuracy: 0.9461 - val_loss: 0.1738
Epoch 9/9
87/87 ━━━━━━━━━━━━━━━━━━━━ 122s 1s/step - accuracy: 0.9812 - loss: 0.0691 - val_accuracy: 0.9581 - val_loss: 0.1593
```

Figure 4: Accuracy when epoch is 9

	precision	recall	f1-score	support
0	0.88	0.78	0.82	9
1	1.00	0.78	0.88	9
2	1.00	1.00	1.00	10
3	1.00	0.78	0.88	9
4	1.00	1.00	1.00	10
5	0.82	1.00	0.90	9
6	1.00	1.00	1.00	7
7	1.00	1.00	1.00	9
8	0.88	1.00	0.93	7
9	0.77	1.00	0.87	10
10	1.00	1.00	1.00	10
11	1.00	1.00	1.00	10
12	1.00	1.00	1.00	10
13	1.00	1.00	1.00	10
14	1.00	1.00	1.00	8
15	1.00	1.00	1.00	9
16	1.00	1.00	1.00	10
17	1.00	1.00	1.00	7
18	1.00	1.00	1.00	9
19	1.00	1.00	1.00	10
20	1.00	1.00	1.00	9
21	1.00	1.00	1.00	7
22	1.00	1.00	1.00	10
23	0.91	1.00	0.95	10
24	1.00	1.00	1.00	9
25	1.00	1.00	1.00	10
26	1.00	1.00	1.00	10
27	1.00	0.89	0.94	9
28	0.89	1.00	0.94	8
29	1.00	1.00	1.00	10
30	1.00	1.00	1.00	10
31	1.00	0.70	0.82	10
32	1.00	1.00	1.00	10
33	1.00	1.00	1.00	10
34	1.00	1.00	1.00	10
35	0.91	1.00	0.95	10
accuracy			0.97	334
macro avg	0.97	0.97	0.97	334
weighted avg	0.97	0.97	0.97	334

Figure 5: Precision, recall, f1-score, support

V. DISCUSSION

A. CNN Model Parameters Proposed

The Adam optimizer is used to train the model, which has learning rate of 0.001. Batch size of 32 is used and the number of epochs are 9. The model is trained with a training set of various varieties of Fruits and Vegetables Images and accuracy evaluation is done by using the test set.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue VI June 2024- Available at www.ijraset.com

B. Impact of Hyper-Parameters on the Model

Batch SizeThe batch size specification affects the accuracy of the classification model. Larger batches impact performance overall and demand more training time. They also consume more memory. Thus, choosing the right batch size is essential to raising the model's quality. In the current study, batch sizes of 8, 16, 32, and 64 are assessed. The accuracy of the model becomes better as the batch size increases from 8 to 16. After falling to 32, it lowers 64 more times. Figure 6 shows that a batch size of 16 individuals produced the best results for the model.

1) Number of Epochs

Epochs are a representation of the number of runs over the whole training dataset. The suggested model goes through 9 training epochs. The model is trained using the batch size of 32, learning rate of 0.0001, and Adam optimizer at various epoch counts: 5, 15, and 20. A classification accuracy of 98.13% is attained in 9 epochs. Figure 7 illustrates how the number of epochs has an impact.

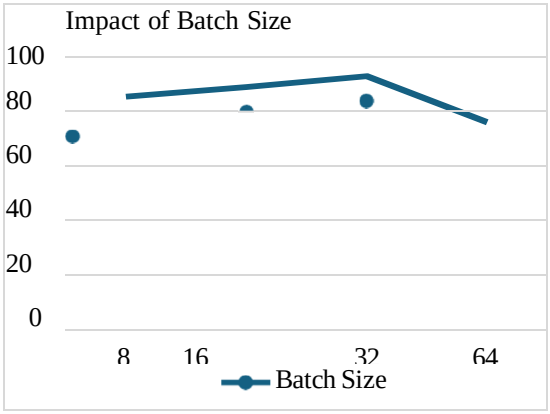


Figure 6: Impact of Epochs

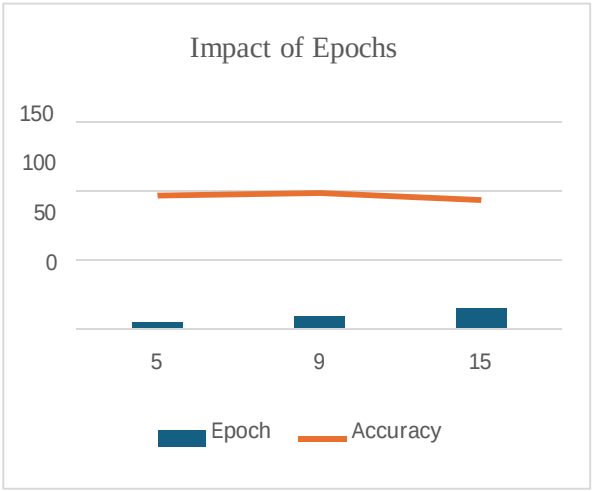


Figure 7: Impact of batch size

2) Optimizers

Another important factor is to choose optimizer which optimizes model's performance. It reduces the loss function by updating the weight parameter. By changing the network's parameters, our objective is to reduce the loss of neural networks. The neural network loss function is assessed by comparing the real and predicted values. Assessment of four optimizers and accuracy comparison is done to figure out the best optimizer. Four optimizers used in presented work are stochastic gradient descent (SGD), Adam, Adagrad, and RMSprop. Based on analysis best optimizer is found to be Adam optimizer which gives the accuracy of 98.33% where as RMSprop , SGD and Adagrad gives the accuracy of 89.19 %, 24.43% and 62.65% respectively. Figure 8 shows the impact of different optimizers.

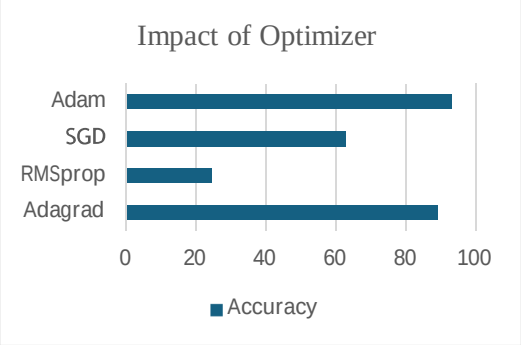


Figure 8: Impact of Optimizer

3) Learning Rates

Neural network training involves updating a particular number of weights. We call this the pace of learning. CNN models' performance is significantly impacted by the learning rate. It can be in the range of 0.0 and 1.0. After training our model with four distinct learning rates, we examined the effects on accuracy. Four different learning rates—0.1, 0.01, 0.001, and 0.0001—were used in this investigation. According to our study, if the learning rate is lowered from 0.1 to 0.0001, the training accuracy increases from 19.97% to 98.33%. Figure 9 displays how learning rates affect outcomes.

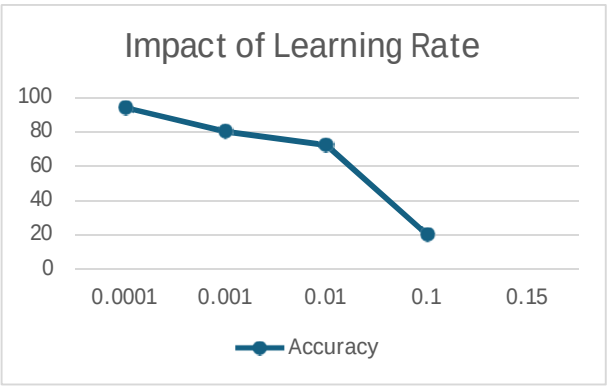


Figure 9: Impact of Learning Rate

C. *The impact of non-linearities and various types of shortcut (residual) connections*

With the removal of ReLU6 at the output of each bottleneck module, accuracy is improved. With shortcut between bottlenecks, it outperforms shortcut between expansions and the one without any residual connections.

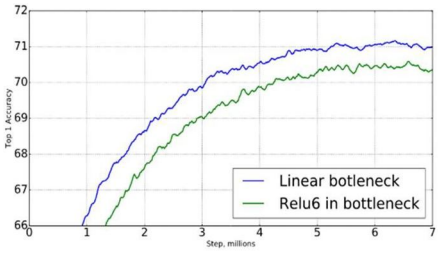


Figure10: Impact of bottleneck

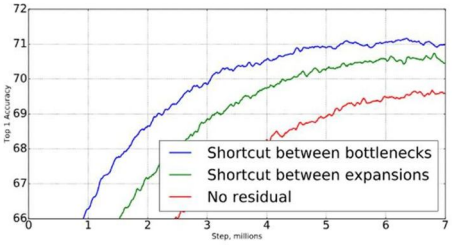


Figure11: Impact of Shortcut

D. Evaluation of Suggested Model's Classification Accuracy And Transfer Learning Models

The optimal set of hyperparameters was obtained by hypertuning the parameters during the training phase. The optimal combinations are batch size: 32, number of epochs: 9, optimizer: Adam, and learning rate: 0.0001. Testing was done on the test dataset once all of these hyperparameters were configured, and the accuracy resulted in 93.17%. VGG16, with an accuracy score of 90.81 %, performs better than any other transfer learning model but not up to with the MobileNetV2.

The suggested model consumes less memory and computes faster because it has fewer filters and parameters. It may thus be used to categorize fruits and vegetables based on their images. The recommended CNN model has a high accuracy of 98.33% due to the combination of convolution and pooling layers and the accurate optimization of hyperparameters. Between the convolution and max pool layers, the RELU activation function is used to improve training while reducing overfitting. There is a 0.5 dropout value. With Adam's optimizer, the error loss is minimized. A comparison of the pre-trained model's performance and the recommended method is shown in Figure 12. The outcomes demonstrate that the recommended strategy outperforms the other transfer learning model in terms of efficiency. The accuracies of MobileNetV2 and other transfer learning models are compared in Table 2.

Table 2:Accuracies of Transfer Learning Models

Model	Accuracy
VGG-16	90.81
VGG-19	76.48
LeNet-5	82.93
AlexNet	83.56
MobileNetV2	98.33

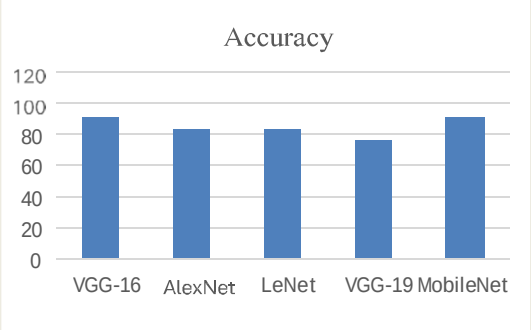


Figure 10:Comparative Study of Accuracies of the proposed method of Pre -Trained models

Confusion Matrix

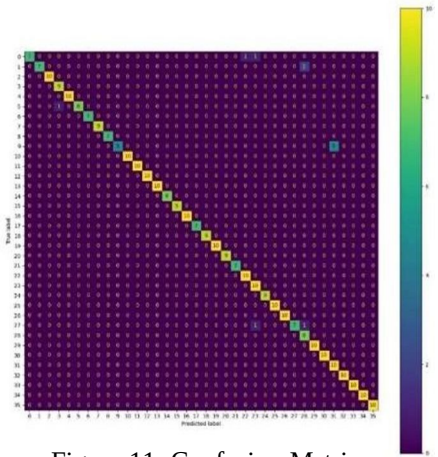


Figure 11: Confusion Matrix

VI. CONCLUSION

In this project, we aimed to develop an efficient and accurate model for classifying fruits and vegetables using the MobileNetV2 pretrained Convolutional Neural Network (CNN).

Our primary goal was to leverage transfer learning to achieve high accuracy on our custom dataset of fruits and vegetables.

We successfully fine-tuned the MobileNetV2 model, achieving an impressive accuracy of 98.33% on the test dataset. This result not only surpasses our initial expectations but also demonstrates the effectiveness of using a lightweight yet powerful pretrained model like MobileNetV2 for image classification tasks.

Throughout the project, we faced several challenges, including ensuring consistent preprocessing, and optimizing the model's hyperparameters. By employing data augmentation techniques and rigorous training processes, we were able to mitigate these challenges and improve the model's robustness and generalization capability.

Our findings highlight the significant potential of transfer learning in enhancing classification accuracy with limited computational resources. The high accuracy achieved indicates that the model is well-suited for practical applications in fruit and vegetable recognition, which can be beneficial for various industries such as agriculture, retail, and food services.

Despite the success, there are areas for future improvement. Enhancing the dataset with more diverse samples and exploring ensemble methods could further boost the model's performance. Reflecting on our journey, we have gained substantial knowledge in machine learning techniques, particularly in transfer learning and model optimization. The collaborative effort and problem-solving skills honed during this project will undoubtedly contribute to our future endeavors.

In conclusion, the project on fruits and vegetables classification using the MobileNetV2 pretrained CNN model has been a significant achievement. The high accuracy attained underscores the model's capability and opens avenues for its application in real-world scenarios. We extend our gratitude to our mentors and peers for their invaluable support throughout this project.

REFERENCES

[1] Karakaya, D., Ulucan, O., & Turkan, M. (2020). A comparative analysis on fruit freshness classification. In 2019 Innovations in Intelligent Systems and Applications Conference (ASYU) (pp. 1-4). IEEE.

[2] Jana, S., & Parekh, R. (2017, March). Shape-based fruit recognition and classification. In International Conference on Computational Intelligence, Communications, and Business Analytics (pp. 184-196). Springer, Singapore.

[3] Saranya, N., Srinivasan, K., Kumar, S. P., Rukkumani, V., & Ramya, R. (2019, September). Fruit classification using traditional machine learning and deep learning approach. In International Conference on Computational Vision and Bio Inspired Computing (pp. 79-89).

[4] Hou, L., Wu, Q., Sun, Q., Yang, H., & Li, P. (2016, August). Fruit recognition based on convolution neural network. In 2016 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD) (pp. 18-22).

[5] IEEE Azizah, L. M. R., Umayah, S. F., Riyadi, S., Damarjati, C., & Utama, N. A. (2017, November). Deep learning implementation using convolutionalneural network in mangosteen surface defect detection. In 2017 7th IEEE International Conference on Control System, Computing and Engineering (ICCSCE) (pp. 242246). IEEE.

[6] Tahir, M. W., Zaidi, N. A., Rao, A. A., Blank, R., Vellekoop, M. J., & Lang, W. (2018). A fungus spores dataset and a convolutional neural network based approach for fungus detection. IEEE transactions on nanobioscience, 7(3), 281-290.

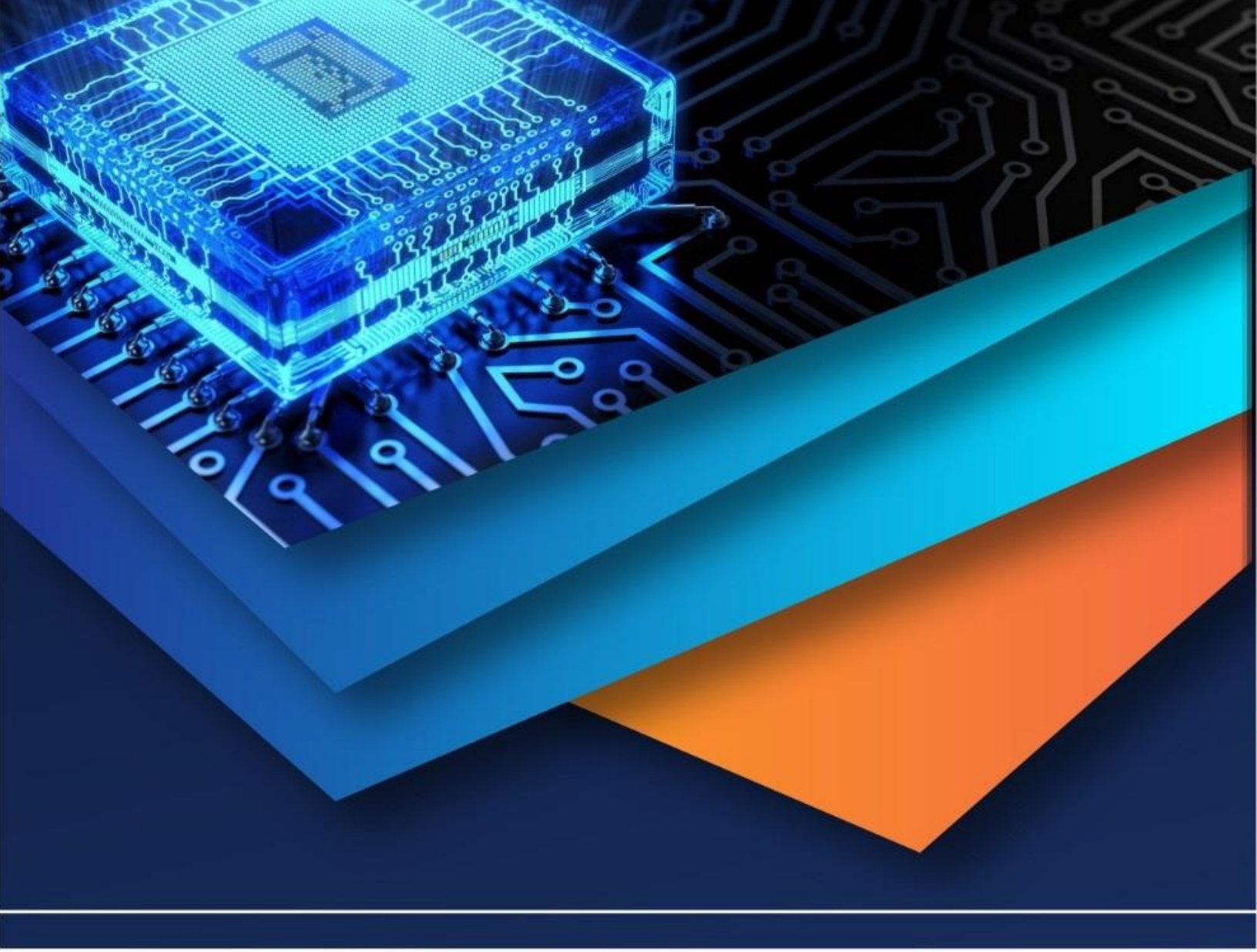
[7] Thenmozhi, K., & Reddy, U. S. (2019). Crop pest classification based on deep convolutional neural network and transfer learning. Computers and Electronics in Agriculture, 164, 104906. Kamilaris, A., & Prenafeta-Boldú, F. X. (2018). Deep learning in agriculture: A survey. Computers and electronics in agriculture, 147, 70-90.

[8] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems, 25, 1097-1105.

[9] LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278-

[10] 2324.

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



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY