Food image recognition based on MobileNetV2 using support vector machine

Sapna Yadav*, Satish Chand*

Jawaharlal Nehru University, New Delhi-110067, India

*Corresponding author

doi: https://doi.org/10.21467/proceedings.114.27

Abstract

The rapid growth in deep learning has made convolutional neural networks deeper and more complex to realize higher accuracy. But many day-to-day recognition tasks need be performed in a limited computational platform. One of the applications is food image recognition which is very helpful in individual's health monitoring, dietary assessment, nutrition analysis etc. This task needs small convolutional neural network based engine to do computations fast and accurate. MoblieNetV2 being simple and smaller in size can incorporate easily into small end devices. In this paper, MobileNetV2 and support vector machine are used to classify the food images. Simulation results show that the features extracted from *Conv_1* layer, *out_relu* layer and *Conv_1_bn* layer of MobileNetV2 and classified using Support Vector Machine have achieved classification accuracies of 84.0%, 87.27% and 83.60% respectively. Because of fewer parameters, smaller size and lesser training time, MobileNetV2 is an excellent choice for real-life recognition tasks.

Keywords: Convolution Neural Network, Image Classification, Machine Learning, MobileNetV2, Deep Learning, Support Vector Machine.

1 Introduction

Two-third of deaths worldwide are due to increased obesity causing diseases like cardiovascular and diabetes. The American Medical Association even classified obesity as a disease in 2013 [1, 2]. An effective way to control obesity is to accurately monitor daily food intake and nutrition amount [3]. Assessment of daily food intake and nutrition amount is a key challenge. A daily dietary monitoring system requires a food recognition system based on deep features. This helps in the medical treatment of chronic disease patients [4]. Food image recognition is a challenging task due to deformable and large visual variations in food images. With traditional methods, it is challenging to recognize food images with high accuracy. Traditional methods for assessing daily food intake are based on experience and human visual recognition, which are very much prone to errors. Recent advancements in computer vision technologies have made promising results to address food recognition problems. Handheld devices such as smartphones can capture food images and automatically acquire accurate diet assessment data. Computer vision systems mainly use two main methods for food image recognition tasks: one uses classical approach and the other applies deep learning based approach. Classical approaches use handcrafted features such as texture and color features [5, 6] with the help of traditional machine learning algorithms such as k-Nearest Neighbor, decision trees [7], artificial neural networks [8] etc. Conventional approaches classify as follows:

- Pre-processes the acquired food image
- Segments the food image part by separating the image background
- Extracting the features and classifying the type of food image using the conventional algorithm.

Ever since AlexNet [9] won the ImageNet challenge, deep learning approaches became popular in computer vision. [10]. The tendency has been to make deeper and more complicated networks with the

Proceedings DOI: 10.21467/proceedings.114; Series: AIJR Proceedings; ISSN: 2582-3922; ISBN: 978-81-947843-8-8

aim to attain higher accuracy [11, 12, 13]. But the main disadvantage is the high complexity of resulting networks.

A lot of real-world applications need to accomplish without computational delay with limited resources. MobileNetV2, which is a variant of MobileNet deep learning architecture, is specially designed for edge devices. MobileNetV2 architecture has small size and fewer parameters still it achieves significant accuracy in food image recognition tasks. This paper uses MobileNetV2 as a pre-trained convolutional neural network trained on the ImageNet dataset. The model extracts complex features of food images from its different intermediate layers and classifies the food images with the help of support vector machine (SVM). Fig. 1 shows the samples of food image samples from the food-101 dataset, which are to be classified.

(a1) (a2) (a3)

Fig. 1 Samples of food images: (a1) Cup cake (a2) French fries (a3) Omelette.

Depthwise separable convolution is the major building block of MobileNets, which decreases the computation in the initial layers [14]. The rest of the paper is organized as follows. Section 2 discusses the methodology used in food image recognition. In Section 3, experimental results are shown. At last, section 4 draws the conclusion from experimental results.

2 Methodology

This section describes the proposed system modules to recognize food labels of the food image dataset. Fig. 2 shows the proposed framework for food image classification.

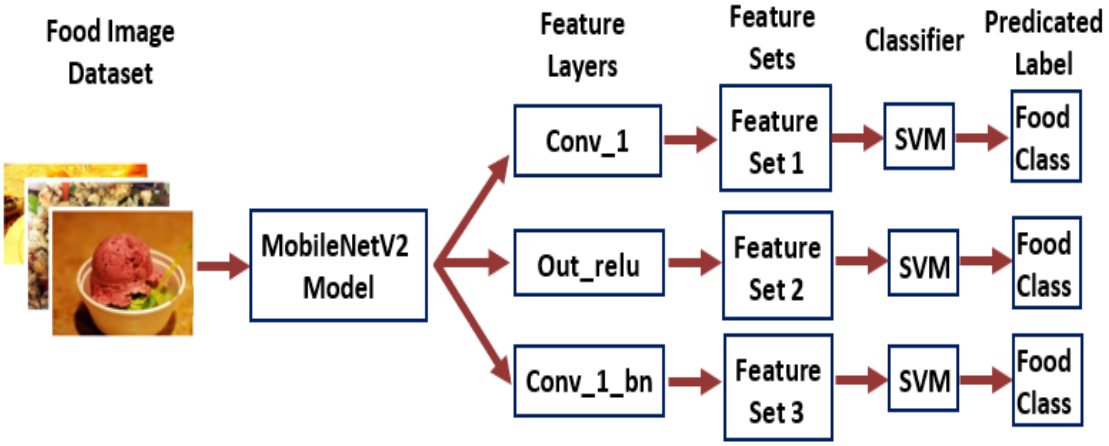

Fig. 2 Proposed framework of MobileNetV2 and SVM for food image classification.

2.1 Dataset

The food image dataset contains 10 food categories taken from the food-101 dataset. Each of the 10 food categories includes 500 images. So, a total of 5000 food images are used in this classification task. In this dataset, 70% of the images are used for training the model, and 30% of them are used for testing purposes.

- *Data Augmentation:* The effectiveness of the dataset can be improved by using some data augmentation techniques. However, contrary to large models, small models are less prone to overfitting. Therefore, this work uses only basic data augmentation methods.

2.1.1 MobileNetV2 Architecture

MobileNetV2 network is 53 layers deep with 3.5 million parameters. The size of the network is 13 MB and it takes input images of size *224x224x3* [15]. Details about core layers, depthwise separable filters and shrinking hyperparameters of the MobileNetV2 network are described below.

1) Layered Architecture: MobileNetV2 is very similar to the ordinary CNN models. However, it uses three innovative structures: a) Depthwise separable convolution b) Linear bottleneck c) Inverted residual.

a) Depthwise Separable Convolution

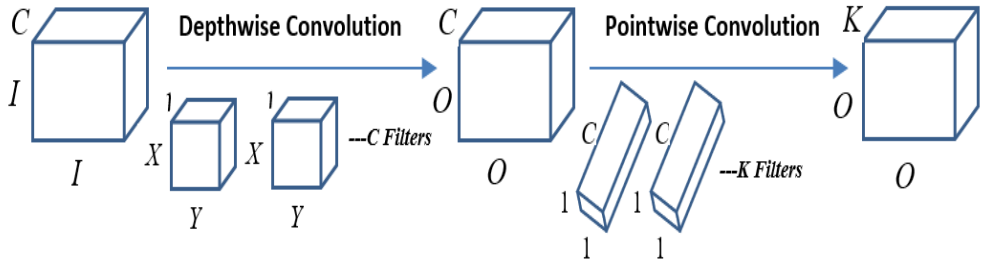

Fig. 3 Depthwise separable convolution in MobileNetV2

These CNNs have lesser number of parameters which reduces overfitting. Also, these are computationally cheaper because of fewer computations. The process of depthwise separable convolution is divided into two parts:

(i) Depthwise convolutions: Depthwise convolution utilizes one filter with each channel. It has a computation cost of:

$$IxIxCxXxY \quad (1)$$

(ii) Pointwise convolutions: Pointwise convolution applies several 1x1 filters to all channels of the output of the previous phase. It has a computational cost of:

$$CxKxIxI \quad (2)$$

$$Depthwise\ separable\ convolution\ cost = Depthwise\ convolution\ cost + Pointwise\ convolution\ cost$$

i.e., Standard convolution has a computation cost of:

$$IxIxCxKxXxY \quad (3)$$

Reduction in computation cost can be calculated by comparing the cost of standard convolution and depth-wise separable convolution. There is quite a good reduction in computing cost at only a small reduction in accuracy [16].

b) Linear Bottleneck: The MobileNetV2 combines the pointwise convolution and bottleneck together and uses pointwise convolution to realize bottleneck. Lots of information is lost when ReLU applies to the data after dimensionality reduction. So linear activation is used in the bottleneck to reduce the loss of information.

c) Inverted Residual: These blocks follow a narrow-wide-narrow approach. It uses 1x1 convolution, then 3x3 convolution, which reduces the number of parameters and then again 1x1 convolution thereby reducing the number of channels. Shortcuts are used directly between the bottlenecks [17].

2) MobileNetV2 Hyperparameters: MobileNetV2 uses shrinking hyperparameters for the cases, requiring the model to be much smaller with high computation.

- *Shrinking Hyperparameters: Width multiplier and Resolution multiplier*

Width multiplier Λ where $\Lambda \epsilon (0,1]$ narrow downs the network evenly at each layer. $\Lambda < 1$ reduces MobileNet, which reduces the computational cost and the number of parameters quadratically. To reduce computational cost another hyperparameter resolution multiplier β is used where $\beta \epsilon (0,1]$. $\beta < 1$ is reduced computational MobileNet and it reduces the computational cost by roughly β^2 [18].

2.1.2 Food image Classification by MobileNetV2 and SVM

Deep features are extracted from different layers of MobileNetv2. Instead of classifying them with the softmax layer, SVM is used to classify these extracted features. Food image dataset is feed into MobileNetv2 and features are extracted by this network after applying depthwise separable convolution, batch normalization and ReLU as non-linearity is applied with pooling operations throughout the network. After that, deep features are extracted from the *Conv_1* layer, *out_relu* layer and *Conv_1_bn* layer of MobileNetv2. These extracted deep features are then fed separately into SVM for the type of food image classification. Multiclass error-correcting codes model using SVM with one-vs-all multiclass classification approach is used.

In the One-vs-all classification SVM, there is one binary SVM for each class to separate that class members from members of other classes [19]. The binary classifier model calculates the scores on the basis of the probability of belonging to concerned classes. These scores are then analyzed and the class is predicted with the class index of the highest probability score [20, 21]

3 Experimental Results

In this paper, we have examined the performance of the MobileNetV2 [22] CNN model with the SVM [23] approach for food image classification. The experimental results are implemented using MATLAB. Model is run on an intel core i7 9th Gen with 2.60 GHz processor, NVIDIA GeForce GTX GPU, 8.00 GB RAM on Windows 10, 64-bit Operating system.

Model performance is measured in terms of accuracy, precision, recall and F1-score. The performance results of the classifier are discussed further. The model uses one-vs-all as the SVM classifier approach. The performance measures are evaluated as shown in Eqs. (4) to (7).

$$Accuracy = \frac{True\ Positive + True\ Negative}{True\ Positive+False\ Positive+True\ Negative+False\ Negative} \quad (4)$$

$$Precision = \frac{True\ Positive}{True\ Positive+False\ Positive} \quad (5)$$

$$Recall = \frac{True\ Positive}{True\ Positive+False\ Negative} \quad (6)$$

$$F1 - Score = 2\ x\ \frac{Precision*Recall}{Precision+Recall} \quad (7)$$

1) Performance analysis of MobileNetV2 and SVM: Performance analysis based on deep features extracted from *Conv_1* layer, *out_relu* layer and *Conv_1_bn* layer of MobileNetv2 classified using SVM is examined. Fig. 4(a) shows the confusion chart based on experimental results obtained by classifying *Conv_1* layer features.

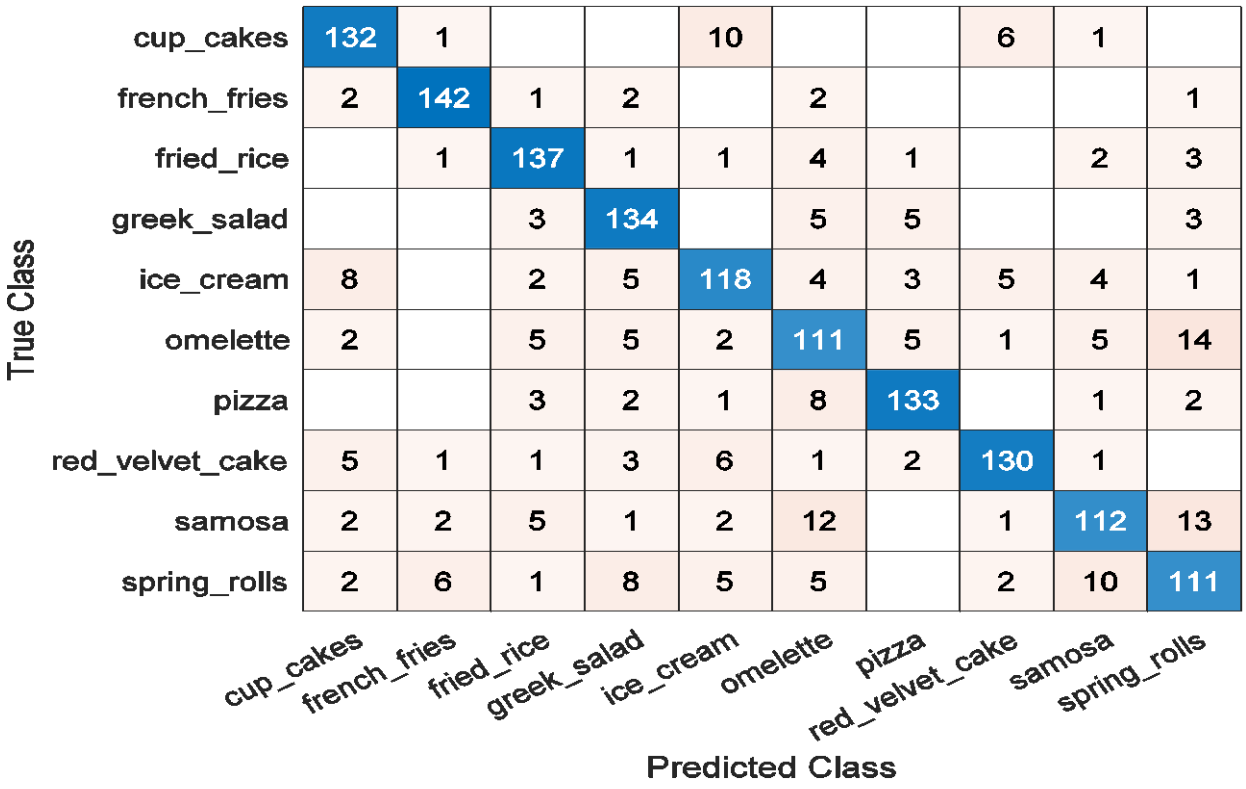

Fig. 4(a) Confusion chart obtained by classifying *Conv_1* layer features

Fig. 4(b) shows the confusion chart based on experimental results obtained by classifying *out_relu* layer features.

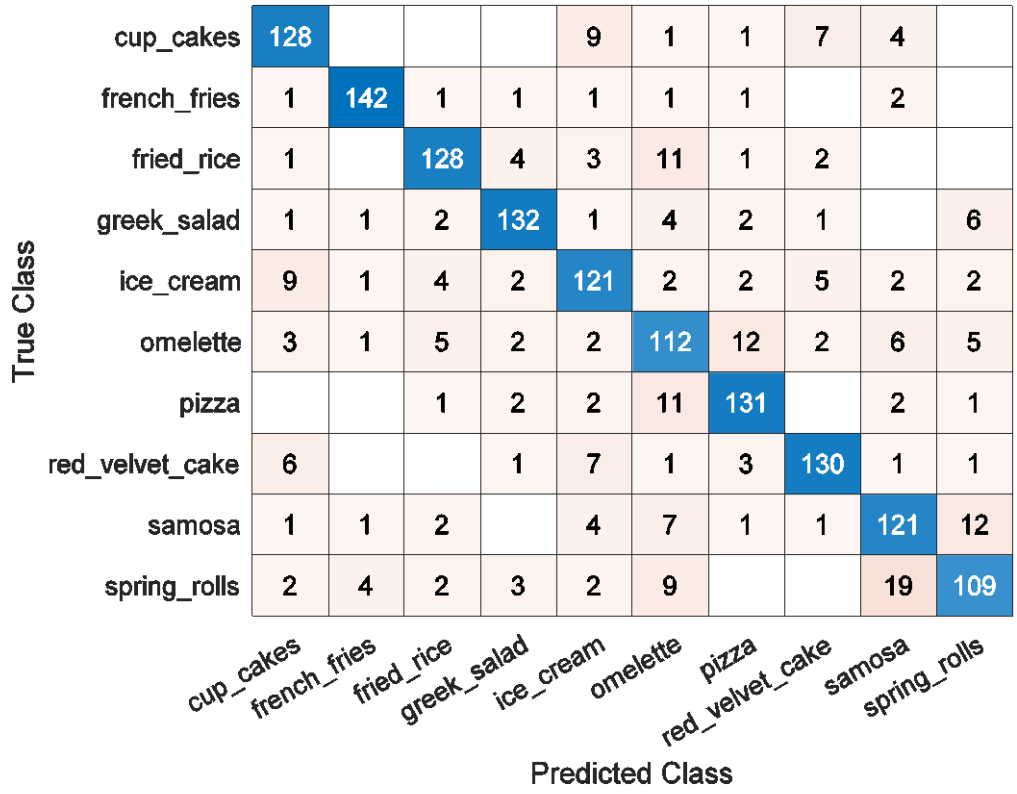

Fig. 4(b) Confusion chart obtained by classifying *out_relu* layer features

Fig. 4(c) shows the confusion chart based on experimental results obtained by classifying *Conv_1_bn* layer features.

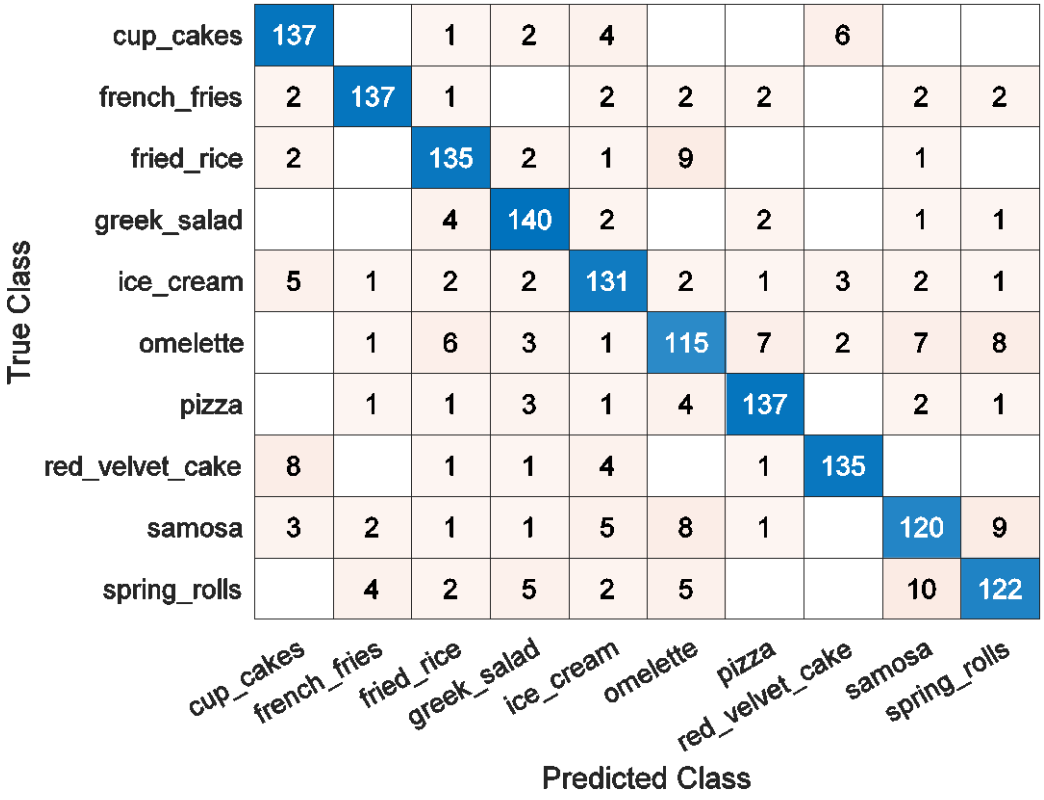

Fig. 4(c) Confusion graph obtained by classifying *Conv_1_bn* layer features

Table 1 shows the MobileNetV2 performance in terms of accuracy, overall precision, overall recall and overall F1-scores achieved at different layers. With the SVM classifier, the model has achieved quite good performance, with *out_relu* layer features performing much better than other layers.

TABLE 1. PERFORMANCE MEASURES OF MOBILENETV2 AND SVM

MobileNetV2 Feature Layers	Accuracy	Overall Precision	Overall Recall	Overall F1_Score
conv_1	84.00%	83.96%	84.00%	75.65%
out_relu	87.26%	87.24%	87.26%	82.43%
convl_bn	83.60%	83.71%	83.60%	74.30%

2) Feature vectors and feature map of MobileNetV2 layers: The number of feature vectors extracted at different layers of MobileNetV2 are shown in Table 2. 62720 different features were extracted at different layers. These feature vectors were fed to SVM classifier for food image classification.

TABLE 2. FEATURE VECTORS AT DIFFERENT LAYERS OF MOBILENETV2

CNN Model	Feature Layer	Feature Vector
MobileNetV2	conv_1	62720
MobileNetV2	out_relu	62720
MobileNetV2	convl_bn	62720

A feature map with 32 channels generated at one of the intermediate layers is shown in Fig. 5.

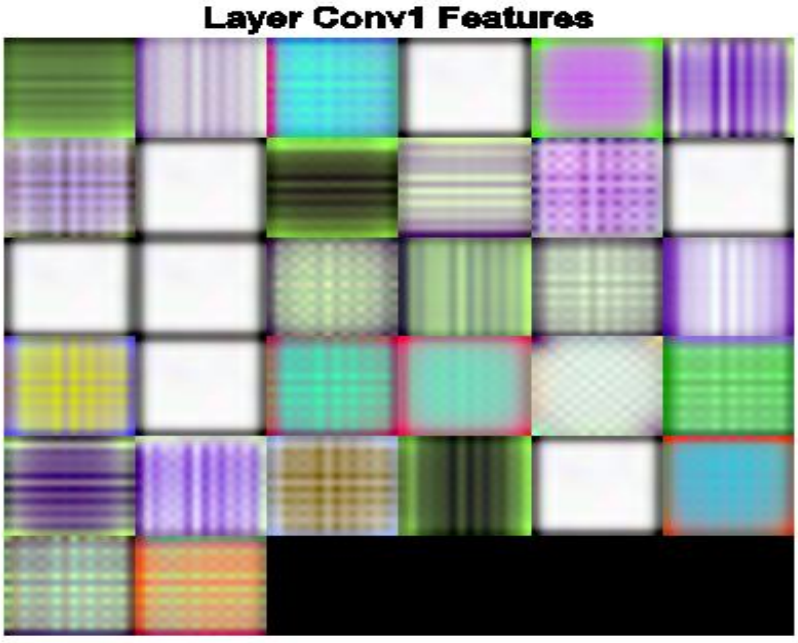

Fig. 5 Feature map at Conv1 layer

Proceedings DOI: 10.21467/proceedings.114
ISBN: 978-81-947843-8-8

Series: AIJR Proceedings
ISSN: 2582-3922

3) Training and testing time comparison: Time taken by the model to train at different layers is shown in Fig. 6. Features trained at the *out_relu* layer have taken more time but have also achieved the highest classification accuracy among the three layers.

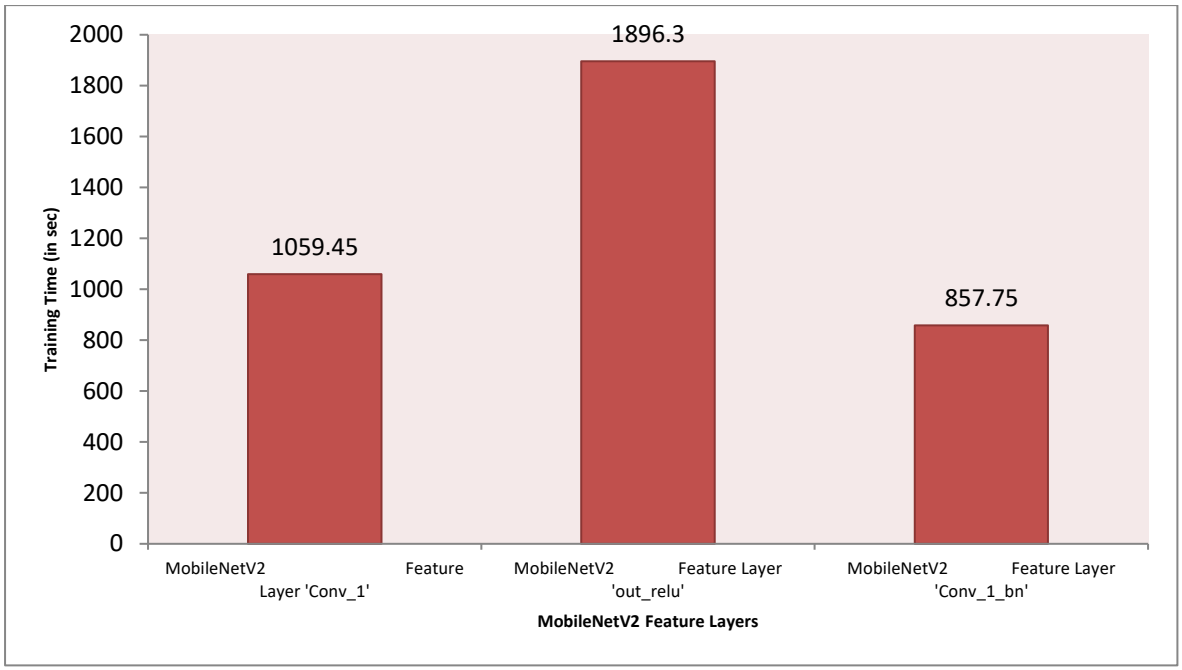

Fig. 6 Training time comparison chart at different layers

Time taken by the model for testing at different layers is shown with a comparison chart in Fig. 7.

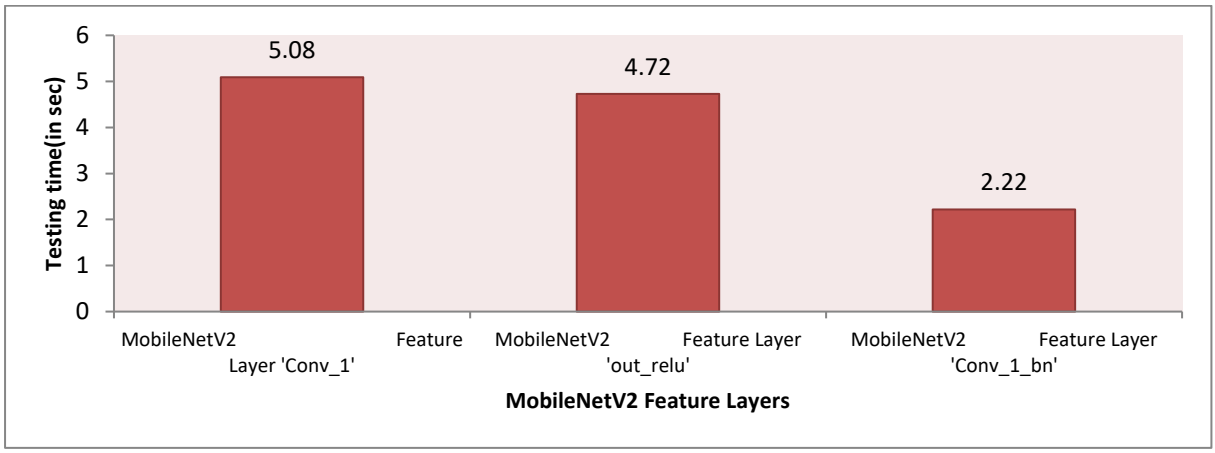

Fig. 7 Testing time comparison chart at different layers

4 Conclusion

In this paper, we evaluated the performance of the MobileNetV2 CNN model in deep learning with the SVM classifier. At first, the deep features of the *Conv_1* layer, *out_relu* layer, *Conv_1_bn* layer were extracted and were then classified by SVM for the type of food image recognition. With this approach, the classification accuracies of 84.00%, 87.26% and 83.60% were achieved at respective layer . MobileNetV2 is a simple and small size network that is easy to integrate with small end devices like smartphones. MobileNetV2 took less time for training and testing without compromising classification accuracy to carry

out recognition tasks in limited time on a computationally limited platform. Therefore, making it an excellent choice for real-life recognition tasks.

References

[1] P. Andrew, "AMA recognizes obesity as a disease," The New York Times. [Online]. [Accessed 12 February 2021].

[2] W. Matthew, "The facts about obesity," American Hospital Association. [Online]. [Accessed March 2021].

[3] R.Z. Franco, R. Fallaize, J.A. Lovegrove, F. Hwang, "Popular Nutrition-Related Mobile Apps: A Feature Assessment," JMIR Mhealth Uhealth, vol. 4, No. 3, doi: 10.2196/mhealth.5846, 2016.

[4] Chua, Z.-Y. Ming, J. Chen, Y. Cao, C. Forde, C.-W. Ngo and T. Seng, "Food Photo Recognition for Dietary Tracking: System and Experiment," in MultiMedia Modeling, Springer International Publishing, pp. 129-141, 2018.

[5] F. Riaz, A. Hassan, R. Nisar, M. Dinis-Ribeiro and M. Tavares Coimbra, "Content-Adaptive Region-Based Color Texture Descriptors for Medical Images," IEEE Journal of Biomedical and Health Informatics, vol. 21, no. doi:10.1109/JBHI.2015.2492464, pp. 162-171, 2017.

[6] G. A. Rahmani, "Efficient Combination of Texture and Color Features in a New Spectral Clustering Method for PolSAR ImageSegmentation," National Academy Science Letters, vol. 40, pp. 117-120, 2017.

[7] J. Xia, P. Ghamisi, N. Yokoya and A. Iwasaki, "Random Forest Ensembles and Extended Multiextinction Profiles for Hyperspectral Image Classification," IEEE Transactions on Geoscience and Remote Sensing, vol. 56, pp. 202-216, 2018.

[8] S. Kaymak, A. Helwan and D. Uzun, "Breast cancer image classification using artificial neural networks," Procedia Computer Science, vol. 120, pp. 126-131, 2017, https://doi.org/10.1016/j.procs.2017.11.219.

[9] A. Krizhevsky, I. Sutskever, G. E Hinton, "Imagenet classification with deep convolutional neural networks," Advances in neural information processing systems, p. 1097–1105, 2012.

[10] O. Russakovsky, H. S. J. Deng, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein and et. al., "Imagenet large scale visual recognition challenge," International Journal of Computer Vision, vol. 115, pp. 211-252, 2015.

[11] Zisserman and K. Simonyan, "Very deep convolutional networks for large-scale image recognition," arXiv preprint arXiv:1409.1556, 2014.

[12] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens and A. Z. Wojna, "Rethinking the inception architecture for computer vision," arXiv preprint arXiv:1512.00567, 2015.

[13] K. He, X. Zhang, S. Ren and A. J. Sun, "Deep residual learning for image recognition," arXiv preprint arXiv:1512.03385, 2015.

[14] J.L. Sifre, "Rigid-motion scattering for image classification," PhD thesis,, 2014. [Online].

[15] "Pretrained Deep Neural Networks," [Online]. Available: https://in.mathworks.com/help/deeplearning/ug/pretrained-convolutional-neural-networks.html. [Accessed March 2021].

[16] L. Kaiser, A. N. Gomez and F. Chollet, "Depthwise Separable Convolutions for Neural Machine Translation," arXiv:1706.03059, 2017.

[17] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov and L.-C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," arXiv:1801.04381, vol. 4, March, 2019.

[18] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto and H. Adam, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," arXiv:1704.0486, vol. 1, April, 2017.

[19] Kai-Bo, DuanJagath, C. Rajapakse, M. N. and Nguyen, "One-Versus-One and One-Versus-All Multiclass SVM-RFE for Gene Selection in Cancer Classification," in Evolutionary Computation,Machine Learning and Data Mining in Bioinformatics, vol. 4447, Berlin, Heidelberg, Springer Berlin Heidelberg, pp. 44-56, 2007.

[20] B. Aisen, "A Comparison of Multiclass SVM Methods," Dec. 2016. [Online]. Available: https://courses.media.mit.edu/2006fall/mas622j/Projects/aisen-project/. [Accessed March 2021].

[21] S. Gupta and S. Amin, "Integer Programming-based Error-Correcting Output Code Design for Robust Classification," arXiv:2011.00144, vol. 1, 2020.

[22] "Help Center," The MathWorks, Inc., [Online]. Available: https://in.mathworks.com/help/deeplearning/ref/mobilenetv2.html. [Accessed March 2021].

[23] "Help Center," The MathWorks, Inc., [Online]. Available: https://in.mathworks.com/help/stats/fitcecoc.html. [Accessed March 2021].