ISSN 2277-2685
IJESR/Jan-Mar. 2025/ Vol-15/Issue-1/181-191
Janni Mounika *et. al.,* / International Journal of Engineering & Science Research

Pre-Trained Deep Neural Network Model of VGG-16 For Flower Image Classification

[1]**Janni Mounika,** mounij131@gmail.com
[2]**M Naresh,** nareshmtech08@gmail.com
[1&2]Newton's Institute of Engineering, Guntur, Andhra Pradesh

Abstract

Flowers are vital to our ecosystem, feeding insects, birds, animals, and humans, and serving as medicinal resources. Understanding flowers helps identify new or rare species, benefiting the medicinal industry. Flower classification is crucial and has been extensively researched. Current methods fall into two categories: manual feature extraction and deep learning. Manual methods extract color, texture, and shape features, combining them with machine learning for classification. However, traditional methods have low accuracy and deep neural networks require large datasets. Our work proposes a fine-tuned VGG16 deep learning model. Experiments on the Oxford flower-102 dataset show that data enhancement improves classification accuracy and robustness, outperforming traditional models.

Keywords: *Deep Neural Network, VGG16, Flower Classification, Convolutional Neural Network*

1. INTRODUCTION

Flowers play a vital role in our ecosystem, contributing to biodiversity, pollinator habitats, and industries like floriculture, cosmetics, and herbal medicine. However, identifying flower species manually is a complex and time-intensive task, even for experts. Flower recognition is crucial for biodiversity protection, as flowers vary widely in color, shape, and petal patterns, making species classification challenging. Automated flower classification is a growing research area in image processing and computer vision. Traditional machine learning approaches rely on handcrafted features like color, shape, and texture but often suffer from low accuracy due to insufficient feature representation. Deep learning, particularly Convolutional Neural Networks (CNNs), has revolutionized image classification by automating feature extraction and achieving superior performance. This project leverages the VGG16 model for flower image classification. VGG16 processes images through convolutional layers using small 3×3 filters with fixed stride and spatial padding to preserve resolution. Five max-pooling layers (2×2 windows, stride 2)
reduce spatial dimensions. The convolutional stack is followed by three fully connected (FC) layers: two with 4096 channels and a final layer with 102 channels, corresponding to flower species classes. A softmax layer produces the output probabilities, and ReLU activation is applied to all hidden layers. By integrating deep learning techniques like VGG16, this approach enhances the speed and accuracy of flower species classification, addressing the challenges posed by complex structures, subtle inter-class differences, and vast class variability in nature.

2. LITERATURE SURVEY

Flower classification has gained significant attention due to its applications in botany, agriculture, and image processing. Researchers have explored various techniques, ranging from traditional feature extraction methods to deep learning-based approaches, to enhance classification accuracy. R. Shaparia et al. [1] presented a flower classification system utilizing texture and color features, demonstrating the effectiveness of handcrafted features in distinguishing different flower types. P. Felzenszwalb et al. [2] introduced a deformable part model trained discriminatively, providing robust performance across multiple scales and deformable parts, which has been influential in object detection tasks, including floral images. Navneet Dalal and Bill Triggs [3] proposed the Histograms of Oriented Gradients (HOG) descriptor, a pioneering method for object detection that has since been applied to flower classification due to its ability to capture edge and gradient structures. Similarly, D.G. Lowe [4] developed the Scale-Invariant Feature Transform (SIFT), which provides robust keypoints and descriptors for image classification. Seeland et al. [5] conducted a comparative study on plant species classification using flower images, evaluating various local feature representations. Their findings highlighted the importance of selecting appropriate feature descriptors for accurate classification. M.-E. Nilsback and A. Zisserman [6] investigated flower segmentation methods to improve classification accuracy, while Y.Y. Boykov and M.P. Jolly [8] introduced interactive graph cuts for optimal boundary and region segmentation, which has been applied to segment floral images effectively. T. Saitoh et al. [7] developed an automatic flower recognition system that leveraged segmentation techniques, demonstrating that accurate boundary identification can significantly improve classification results. M.-E. Nilsback and A. Zisserman [9] proposed a visual vocabulary approach for flower classification, combining local feature descriptors with machine learning techniques. The advent of deep learning has further revolutionized the field, as evidenced by studies such as Yuki Tatsunami and Masato Taki [10], who utilized deep LSTM for image classification. M. R. Banwaskar and A. M. Rajurkar [11] introduced a feature fusion-based flower classification system, integrating multiple features to enhance accuracy. M. Mehdipour Ghazi et al. [12] optimized transfer learning parameters for plant identification using deep neural networks, showcasing the adaptability of pre-trained models in floral classification tasks. M. Tian et al. [13] and S. Giraddi et al. [14] applied deep learning models for flower classification, achieving promising results by leveraging convolutional neural networks (CNNs). I. Gogul and V. S. Kumar [15] demonstrated the effectiveness of transfer learning in CNNs for recognizing different flower species. Sushma L. and K.P. Lakshmi [16] analyzed different CNN models for image classification, providing insights into their performance on floral datasets. Huthaifa Almogdady et al. [17] developed a flower recognition system using neural networks, emphasizing the importance of image preprocessing and network architecture. A. Krizhevsky et al. [18] pioneered the use of deep convolutional neural networks for large-scale image classification, setting a benchmark for subsequent research. Fei Hu et al. [19] focused on salient feature learning using CNNs, improving flower classification performance. Mesut Toġaçar et al. [20] explored feature extraction and selection methods in CNN models for flower species classification, highlighting the role of optimized feature sets. Farhana Sultana et al. [21] discussed advancements in CNN-based image classification, providing a comprehensive overview of the state-of-the-art techniques.

3. MATERIALS AND METHODS

There are five modules divided in this project in order develop the concept of Flower Prediction using VGG-16

Algorithms. They are listed below

1. IMAGE Data Preprocessing
2. Feature Engineering
3. Train test Splitting
4. Model Creation
5. Model Evaluation

i) Data Pre-processing: Image pre-processing is crucial for noise removal and enhancing image quality. By refining raw flower images and removing irrelevant background elements, preprocessing significantly improves the clarity and accuracy of subsequent analysis, provided the methods are carefully chosen.

ii) Image enhancement: Image enhancement improves the visual quality of an image, providing a better representation for subsequent detection phases. It is typically categorized into three types.

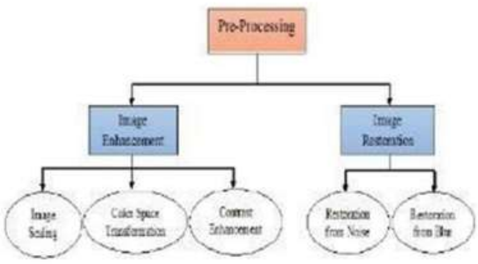

Fig.1: Image Pre-processing

iii) Image Scaling: To address size inconsistencies in flower images from various sources, scaling is applied. This process standardizes the image dimensions by fixing the height and width in pixels.

iv) Colour Space Transformation: Colour plays a key role in flower identification and classification, making accurate color representation essential for processing. Common color spaces include RGB, HSV, HSI, LAB, and YCbCr, with RGB being widely used. However, RGB has limitations, leading to the use of other spaces.

Fig.2: (a) RGB image (b) Gray Scale image (c) Binary image

For flower images, transformations to grayscale and binary formats enhance petal edge detection and intensity analysis. This process refines the image for further processing. Fig.2 illustrates gray scale and binary transformations using MATLAB**.**

v) Contrast Enhancement: Contrast enhancement improves image quality by highlighting the clarity difference between the foreground and background, refining boundaries, and enhancing accuracy. It plays a vital

ISSN 2277-2685
IJESR/Jan-Mar. 2025/ Vol-15/Issue-1/181-191
Janni Mounika *et. al.,* / International Journal of Engineering & Science Research

role in flower image processing. Techniques are categorized into linear and non-linear methods. Figure 3 shows linear contrast enhancement approaches.

Fig.3: Linear Contrast Enhancement Methods

vi) Linear and Non-Linear Contrast Enhancement: In linear contrast enhancement, methods like contrast stretching adjust gray-level values to extend the image histogram across the full range, improving image contrast. Fig 5 shows results of linear techniques applied to flower images.

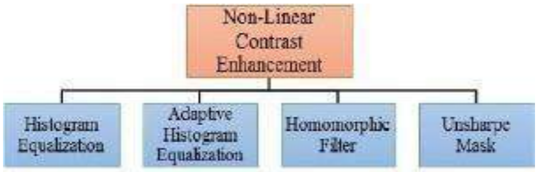

Fig.4: Non-Linear contrast Enhancement Methods

Non-linear contrast enhancement focuses on histogram equalization and algorithms, mapping input values to various output values. However, it may reduce the brightness of flower images. Non-linear enhancement techniques are also illustrated in Figure 4.

vii) Enhancement Techniques for Flower Identification: In flower identification, detailed features are more critical than global ones. Effective enhancement methods include Adaptive Histogram Equalization (AHE), Unsharp Masking, and Histogram Equalization (HE). Figure 5 shows the results of applying these non-linear techniques to flower images.

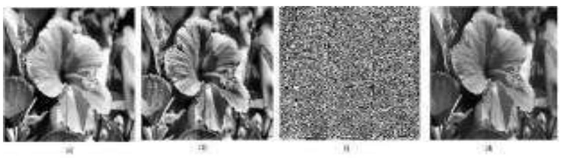

Fig.5: (a) Histogram Equalization (b) Adaptive Histogram Equalization (c) Holomorphic Filter (d) Unsharp Masking

viii) Image Restoration from Noise: Image restoration is crucial for preprocessing flower images, especially for handling various types of noise. Effective denoising must eliminate noise while preserving edges. Noise types include:

1. **Gaussian Noise**: Caused by sensor issues, low light, high temperatures, or transmission errors, also known as amplifier noise.
2. **Salt & Pepper Noise**: Independent, uncorrelated "spike" noise appearing as black or white spots on images.

3. **Poisson Noise**: Photon noise affecting signals in varying proportions.
4. **Speckle Noise**: Random variations in signals reflected from objects.

Noise removal methods are broadly classified into Spatial Filtering and Transform Domain Filtering. Common spatial filters include Mean, Median, Wiener, and Total Variation filters. Fig 6 shows restored images processed using an image processing toolkit.

Fig.6: (a) Gaussian noise (b) Salt and Pepper noise (c) Speckle noise (d) Poison noise

ix) Noise Removal Filters

- Arithmetic Mean: Simplest mean filter, effective against Gaussian noise but can blur edges.
- Geometric Mean: Better at preserving image details than arithmetic mean.
- Harmonic Mean: Works well with salt and Gaussian noise but not pepper noise.
- Adaptive Local Noise Reduction: Effective for random noises like speckle and Gaussian.
- Adaptive Median Filter: Preserves details while smoothing non-impulse noise.
- Min/Max Filters: Identify the darkest/brightest points in an image.
- Median Filter: Removes outliers like salt and pepper noise without blurring edges.
- Mid-Point Filter: Best for random distributed noises like speckle noise.

x) Transform Domain Filtering

Relies on wavelet transforms, an extension of Fourier transforms, to analyse data at different resolutions. Techniques include Visu Shrink, Sure Shrink, Bayes Shrink, Neigh Shrink, Oracle Shrink, Smooth Shrink, and LAWML. Wavelets are mathematical functions that enhance noise reduction with precision.

xi) Image Restoration from Blur

Blur occurs due to poor focus or motion between the camera and the subject. Key deblurring methods include:

- **Wiener Filter**: Commonly used for deblurring and noise reduction.
- **Inverse Filter**: Restores images by reversing blur effects.
- **Neural Network Approach**: Highly effective in agriculture applications, also removes noise.
- **Lucy-Richardson Algorithm**: Utilized for iterative image restoration. In MATLAB 2017, the Wiener filter has been applied for deblurring, and the results are shown in Fig. 7.

Fig.7: (a) Blur image (b) Restoration of Blurred image (c) Restoration of Blurred using Wiener filter

Xii) Feature Engineering: Feature engineering transforms raw data into meaningful features for predictive modelling in machine learning. This process enhances model performance by preparing data effectively.

xiii) Train-Test Split

The train-test split technique evaluates the performance of machine learning models for classification or regression.

- Training Dataset: Used to train the model.
- Test Dataset: Evaluates the model's accuracy on unseen data.

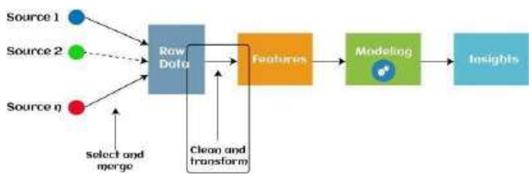

Fig.8: Train Test Splitting

This approach simulates real-world scenarios by testing the model on new examples, ensuring its ability to generalize beyond the training data.

xiv) Model Creation

The final step in machine learning is deploying the model in a live environment to handle unseen data and deliver actionable outcomes. This phase validates the return on investment by enabling the model to perform its trained tasks.

In this approach, the **VGG pretrained model** was utilized for image classification.

VGG (Visual Geometry Group): A deep CNN architecture with small convolutional filters, offering two versions **VGG-16** and **VGG-19**, with 16 and 19 convolutional layers respectively.

Key Features:

- Trained on Nvidia Titan Black GPUs.
- Classifies images into 1,000 object categories.
- Supports an image input size of 224x224 pixels.
- Includes 16 convolutional layers and 3 fully connected layers in the VGG-16 architecture.

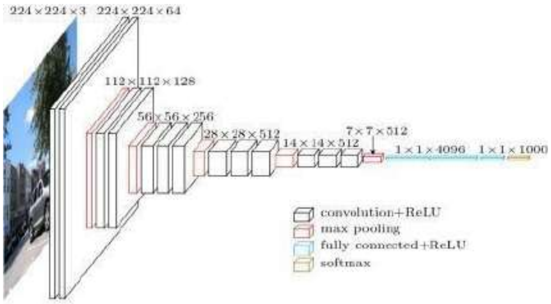

Fig.9: The Architecture VGG model

xv) Training and Model Architecture: The training process uses a pre-trained VGG network, followed by defining a new untrained feed-forward classifier with ReLU activations and dropout. The classifier layers are trained on extracted features, tracking validation loss and accuracy to tune hyperparameters.

Input and Initial Layers

The network processes a fixed 224x224 RGB image:

- **First Stack**: Two convolutional layers (3x3 filters, 64 filters each) with ReLU activations.
- Spatial resolution preserved using 1-pixel padding and 1-pixel stride.
- Max pooling (2x2, stride 2) reduces size to **112x112x64**.
- **Second Stack**: Two convolutional layers with 128 filters.
- Output size becomes **56x56x128**.
- **Third Stack**: Three convolutional layers with 256 filters.
- Output size: **28x28x256**.
- **Fourth & Fifth Stacks**: Three convolutional layers each, with 512 filters.
- Final output size: **7x7x512**.

Fully Connected Layers

- Flattened activations flow into three fully connected layers:
- First two layers: 4,096 neurons each.
- Final layer: 102 neurons for 102 flower classes.
- Softmax activation is applied for classification.

Preprocessing Steps

- **Resizing**: Scale images, maintaining aspect ratio, with the shortest side as 256 pixels. Resize to 224x224 for input.
- **Normalization**: Convert pixel values from 0-255 to 0-1, then normalize using mean [0.485, 0.456, 0.406] and standard deviation [0.229, 0.224, 0.225] for each channel.

Dataset Splits

- **Training Set**: Includes random transformations like scaling, cropping, and flipping to improve generalization.
- **Validation and Testing Sets**: Resize and crop images to 224x224 without additional transformations, measuring performance on unseen data.

Convolutional Neural Network (CNN)

Inspired by biological visual cognition, CNNs consist of input, convolutional, pooling, fully connected, and output layers. Convolutional and pooling layers alternate to process features, extracting patterns from images. The convolution process involves local neuron connections, weighted averages, and non-linear activation, producing feature maps via filters.

1. Convolutional Layer

- Extracts image features like edges, lines, and shapes in initial layers, and more complex features in deeper layers.
- Convolution kernels are optimized through backpropagation.
- Mimics human vision: locally recognizing features, then combining them for global understanding.

2. Pooling Layer

- Also called subsampling, reduces feature map size while preserving key features.
- Prevents overfitting by compressing data without losing important details.
- Types:
- **Max Pooling**: Selects the maximum value in a region.
- **Average Pooling**: Takes the mean value in a region.
- Reduces computation time for convolution operations.

3. Activation Function

- Introduces non-linearity to the network, enabling it to solve complex problems.
- Commonly used: **ReLU** (Rectified Linear Unit)
- Formula: f(x)=max(x,0)f(x) = \max(x, 0)f(x)=max(x,0)
- Ensures gradient flow for x>0x > 0x>0, prevents vanishing gradient issues, and accelerates convergence.

4. Fully Connected Layer

- Located at the end of the network, connects all nodes from the preceding layers.
- Integrates extracted features for classification.
- Converts 2D feature maps into a 1D vector for classification tasks.

5. Softmax Classifier

- Serves as the output layer.
- Converts the final vector into probabilities for each class.
- Outputs a vector where each value is between 0 and 1, representing the likelihood of the sample belonging to a specific class.

4. RESULT

4.1Dataset Description

The Oxford 102 **Flower Dataset** is utilized for experimentation, comprising 8,189 images across 102 flower categories commonly found in the United Kingdom. Each category contains 40 to 258 images, making this dataset more challenging due to:

- Large-scale, pose, and lighting variations.
- Significant intra-category variation.
- Several visually similar categories.

Figure 10 illustrates the class labels for all 102 flower types.

Fig.10: The 102 flower dataset

4.2 Empirical Evaluations

The VGG16 pretrained model was used for image classification on the 102 Flower Dataset. The experiment was conducted over 3 epochs, and the results are summarized in Table 1.

Evaluation Measure	Image Classification
Validation Accuracy	0.851
Test Accuracy	0.762
Validation loss	0.709

Table 1: The accuracies of proposed approach for image classification

In Table1, the proposed approach achieved a validation accuracy of 0.851 and a test accuracy of 0.762 for the 102 flower image classification. Additionally, it recorded a low validation loss of 0.709.

5. CONCLUSION

This work utilized the VGG16 model to train an image classifier for flower species recognition. By fine-tuning the pre-trained VGG16 model, the approach leveraged transfer learning to differentiate flower types effectively. The proposed model achieved a validation accuracy of 0.851, test accuracy of 0.762, and a validation loss of 0.709 for the 102 flower image classification.

REFERENCES

[1] R.Shaparia, N. Patel, and Z.Shah, “Flower classification using texture and color features”, 2: p. 113-118, 2017.

[2] P. Felzenszwalb, D. McAllester, D. Ramaman, “A discriminatively trained, multiscale, deformable part model”, Proceedings of the IEEE CVPR 2008.

[3] Navneet Dalal and Bill Triggs, “Histograms of oriented gradients for human detection”, Proceedings of the IEEE CVPR 2005.

[4] DG Lowe, “Distinctive image features from scale-invariant keypoints”, International Journal of Computer Vision, 2004.

[5] Seeland M, Rzanny M, Alaqraa N, Wäldchen J, Mäder P, “Plant species classification using flower images—A comparative study of local feature representations”. PLoS ONE 12(2): e0170629. doi:10.1371/journal.pone.0170629

[6] M.-E. Nilsback and A. Zisserman. Delving into the whorl of flower segmentation. In Proc. BMVC., volume 1, pages 570–579, 2007.

[7] T. Saitoh, K. Aoki, and T. Kaneko. Automatic recognition of blooming flowers. In Proc. ICPR, volume 1, pages 27–30, 2004.

[8] Y. Y. Boykov and M. P. Jolly. Interactive graph cuts for optimal boundary and region segmentation of objects in ND images. In Proc. ICCV, volume 2, pages 105–112, 2001.

[9] M.-E. Nilsback and A. Zisserman. A visual vocabulary for flower classification. In Proc. CVPR, volume 2, pages 1447–1454, 2006.

[10] Yuki Tatsunami, Masato Taki, “Sequencer: Deep LSTM for Image Classification”, 36th Conference on Neural Information Processing Systems (NeurIPS 2022), arXiv:2205.01972v4 [cs.CV] 12 Jan 2023

[11] [11] M. R. Banwaskar, A. M. Rajurkar, “An Efficient Flower Classification System using Feature Fusion”, International Journal of Engineering Trends and Technology, Volume 70 Issue 11, 70-80, November 2022

[12] M. Mehdipour Ghazi, B. Yanikoglu, and E. Aptoula, "Plant Identification Using Deep Neural Networks via Optimization of Transfer Learning Parameters," Neurocomputing, vol. 235, pp. 228-235, 2017.

[13] M. Tian, H. Chen, and Q. Wang, "Flower Identification Based on Deep Learning," Journal of Physics: Conference Series, vol. 1237, no. 2, 2019.

[14] Giraddi, S. Seeri, P. S. Hiremath and J. G.N, "Flower Classification using Deep Learning Models," International Conference on Smart Technologies in Computing, Electrical and Electronics, pp. 130-133, 2020.

[15] Gogul and V. S. Kumar, "Flower Species Recognition System using Convolution Neural Networks and Transfer Learning," IEEE Fourth International Conference on Signal Processing, Communication and Networking (ICSCN), pp. 1-6, 2017.

[16] Sushma L, and K.P. Lakshmi, "An Analysis of Convolution Neural Network for Image Classification using Different Models," International Journal of Engineering Research & Technology (IJERT), vol. 9, no. 10, 2020.

[17] Huthaifa Almogdady, Dr. Saher Manaseer, and Dr.Hazem Hiary, "A Flower Recognition System Based on Image Processing and Neural Networks," International Journal of Scientific and Technology Research, vol. 7, no. 11, pp. 166-173, 2018.

[18] Krizhevsky, I. Sutskever, and G.E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," Advances in Neural Information Processing Systems, pp. 1-9, 2012.

[19] Fei Hu, Fuguang Yao, and Changjiu Pu, "Learning Salient Features for Flower Classification Using Convolutional Neural Network," 2020 IEEE International Conference on Artificial Intelligence and Information Systems (ICAIIS), pp. 476-479,2020.

[20] Mesut Togˇaçar a, Burhan Ergen, and Zafer Cömert, "Classification of Flower Species by using Features Extracted from the Intersection of Feature Selection Methods in Convolutional Neural Network Models," Measurement, vol. 158, 2020.

[21] Farhana Sultana, Abu Sufian, and Paramartha Dutta, "Advancements in Image Classification using Convolutional Neural Network," Fourth IEEE International Conference on Research in Computational Intelligence and Communication Networks, pp. 122-129, 2018.

[22] Neda Alipour, Omid Tarkhaneh, Mohammad Awrangjeb, Flower Image Classification Using Deep Convolutional Neural Network, Conference Paper · May 2021 DOI: 10.1109/ICWR51868.2021.9443129

[23] Zhao Jiantao1, Chu Shumin, Research on Flower Image Classification Algorithm Based on Convolutional Neural Network, 2021 International Conference on Big Data and Intelligent Algorithms (BDIA 2021), Journal of Physics: Conference Series 1994 (2021) 012034

[24] Xiaoxue Li, Rongxin Lv, Yanzhen Yin, Kangkang Xin4, Zeyuan Liu and Zhongzhi Li, Flower image classification based on generative adversarial network and transfer learning, IOP Conf. Series: Earth and Environmental Science 647 (2021) 012180