International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 12 | Dec - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

Classification And Storage of Images Using ML

Mr Vedananda D E[1], Yashwanth B[2], Suprith M P[3], Suraj S Atresh[4], Suhail D[5]
Department of Computer Science and Engineering
Jawaharlal Nehru New College of Engineering, Shivamogga, Karnataka, India.
[1]vedanandade@jnnce.ac.in, [2]yashwanthb107@gmail.com, [3]suprith2284@gmail.com,
[4]surajatresh30@gmail.com, [5]suhild434@gmail.com

***Abstract*—The rapid growth in the number of images captured on mobile devices has created challenges in effectively organizing and managing large photo collections. Traditional image sorting methods are largely manual, time-consuming, and inefficient when handling unstructured data at scale. This paper presents an automated image classification and storage system based on a lightweight deep learning architecture, MobileNetV2, designed for efficient deployment in resource-constrained environments. The proposed system analyzes the visual content of images, classifies them into predefined categories, and automatically organizes them into a structured folder hierarchy for easy access and retrieval. By integrating efficient deep learning–based classification with automated storage management, the system significantly reduces user effort, improves image organization, and enables faster access to visual information, making it well suited for mobile and embedded applications.**

***Index Terms*—Image Classification, Image Storage, Machine Learning, Deep Learning, Convolutional Neural Networks (CNNs), Mobile Applications, Automated Organization, Data Management.**

I. Introduction

In the digital age, the exponential growth of image data across various domains—such as healthcare, surveillance, e-commerce, social media and personal images—has created a pressing need for efficient methods of image classification and storage. Traditional techniques often fall short in handling large volumes of unstructured image data with accuracy and speed. Machine Learning (ML) offers powerful tools for automating the classification process by enabling systems to learn patterns and features from vast image datasets without explicit programming. Image classification involves assigning predefined labels to images based on their visual content. ML models, particularly Convolutional Neural Networks (CNNs), have proven highly effective in learning complex features and patterns from images, surpassing traditional methods in terms of accuracy and scalability. Once classified, images can be efficiently stored based on their categories, reducing redundancy, improving retrieval times, and optimizing storage space. At the core of the detection system lies a convolutional neural network (CNN), a class of deep learning models highly effective for image analysis tasks. CNNs automatically learn hierarchical feature representations from the input images, which allows the model to detect complex patterns and textures within the lesions that might indicate malignancy. This capability enables CNNs to outperform many traditional computer vision techniques that rely on manually designed features. This project explores the integration of machine learning techniques for both automated image classification and intelligent storage solutions, aiming to enhance data organization, reduce human effort, and enable faster access to relevant visual information.

II. Literature Survey

The field of image classification has evolved significantly with the advancement of deep learning. Our work is built upon foundational and recent research in CNN architectures, with a particular focus on models suitable for mobile applications.

Alex Krizhevsky et al. [6] proposed AlexNet, a deep convolutional neural network trained on the ImageNet dataset using techniques such as ReLU activation, dropout, data augmentation, and GPU acceleration to achieve high accuracy in large-scale image classification. While the model significantly improved hierarchical feature extraction and classification performance, it suffers from high computational complexity, a large number of parameters, and heavy dependence on large labeled datasets, making it unsuitable for real-time and mobile applications.

Fanello et al. [1] introduced a sparsity-based multi-class image classification approach using dictionary learning, where images are represented through sparse and discriminative features derived from class-specific dictionaries. Although this method improves recognition accuracy in cluttered and occluded scenes, its performance is highly dependent on dictionary quality and feature selection, and it does not scale efficiently for large datasets.

Kaiming He et al. [8] proposed ResNet, a deep CNN architecture that uses residual connections to address vanishing gradient issues and enable effective training of very deep networks. Despite achieving state-of-the-art results on ImageNet, ResNet models are computationally intensive, require high memory usage, and are less suitable for deployment on resource-constrained devices.

A. G. Howard et al. [3] introduced MobileNets, an efficient CNN architecture based on depthwise separable convolutions and controlled using width and resolution multipliers to balance accuracy and latency. While MobileNets are well-suited for mobile and embedded vision tasks, they exhibit a slight reduction in accuracy compared to deeper CNNs,

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 12 | Dec - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

especially for complex classification problems.

Pushparaja Murugan [5] proposed a deep convolutional neural network for multi-class image classification using grayscale images, regularization techniques, and Bayesian optimization for hyperparameter tuning. Although the approach improves generalization and reduces overfitting, it requires careful parameter tuning and large datasets, and the use of grayscale images limits performance for color-sensitive classification tasks.

Mingxing Tan and Quoc V. [7] Le presented EfficientNet, a family of CNNs that use compound scaling to uniformly scale depth, width, and resolution for improved accuracy and efficiency. Despite achieving strong performance with fewer parameters, the architecture relies on neural architecture search, making it complex to design and still computationally demanding for low-end devices.

W. A. Ezat et al. [2] proposed a transfer learning-based CNN approach where pre-trained ImageNet models are fine-tuned on the PASCAL VOC dataset for multi-class image classification. While this method reduces training time and computational cost, its effectiveness depends on the similarity between source and target datasets, and fine-tuning can still require significant resources.

Jialu Zhang et al. [4] introduced a spatial-context-aware deep neural network that incorporates spatial dependencies and contextual information to improve semantic understanding and classification accuracy. However, the increased model complexity leads to higher computational costs, and the spatial context assumptions may not generalize well across diverse image datasets.

III. Design and Implementation

A. System Architecture and Workflow

The overall design of the proposed image classification and storage system follows a structured and modular workflow, as illustrated in Figure 1. The flow diagram presents a clear end-to-end view of the system, starting from image acquisition and ending with classified image storage and result display. Each block in the diagram represents a functional module that contributes to accurate classification and automated organization of images.
The system is designed to handle common image formats such as JPG, PNG, and WEBP, making it suitable for real-world usage scenarios, especially in mobile environments. The modular architecture ensures scalability, ease of maintenance, and smooth integration with an Android application.

B. Image Input Module

As shown in Figure 1, the process begins with the Image Input stage, where the user uploads or selects an image. This module acts as the entry point of the system and validates the selected file to ensure it is a supported image format. Once validated, the image is forwarded to the preprocessing unit for further operations.

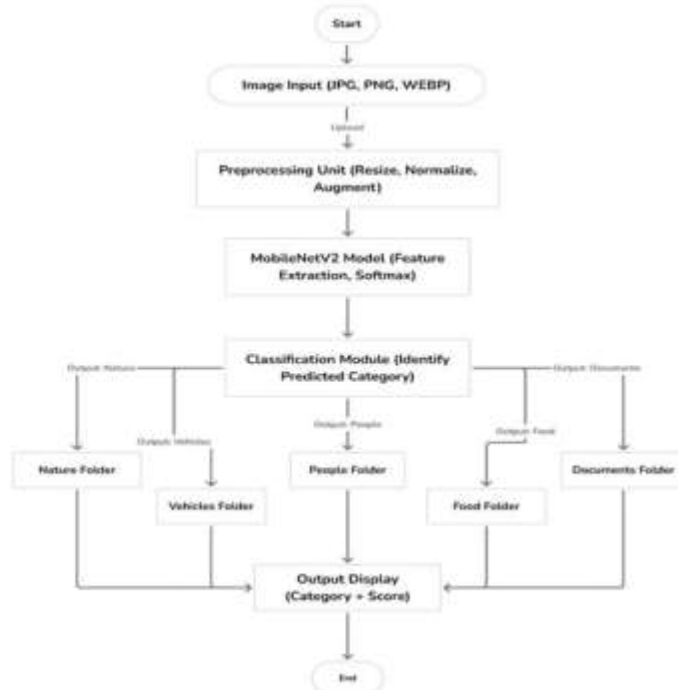

Fig. 1. Workflow of the proposed MobileNetV2-based image classification and automated folder storage system.

C. Preprocessing Unit

The Preprocessing Unit, depicted as the next block in Figure 1, prepares the input image for the deep learning model. The image is resized to 224 × 224 pixels, normalized to scale pixel values between 0 and 1, and augmented during training using transformations such as rotation, zooming, and horizontal flipping. These preprocessing steps improve model robustness, reduce overfitting, and ensure compatibility with the MobileNetV2 architecture.

```
def preprocess_image(img_path):
    img = image.load_img(img_path,
                         target_size=(224, 224))
    img_array = image.img_to_array(img)
    img_array = np.expand_dims(img_array, axis=0)
    img_array = preprocess_input(img_array)
    return img_array
```

D. MobileNetV2 Feature Extraction and Classification

Following preprocessing, the image is passed to the MobileNetV2 model, as highlighted in Figure 1. MobileNetV2 is used due to its lightweight architecture and efficiency, making it suitable for mobile and embedded systems. The model performs hierarchical feature extraction using depthwise separable convolutions. A Softmax layer at the output generates probability scores for each predefined class.

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 12 | Dec - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

```
base_model = MobileNetV2(weights='imagenet',
                         include_top=False, input_shape=(224,
                         224, 3))

x = GlobalAveragePooling2D()(base_model.output) x =
Dense(128, activation='relu')(x)
output = Dense(5, activation='softmax')(x)

model = Model(inputs=base_model.input,
              outputs=output)
```

E. Classification Module and Category Identification

The Classification Module identifies the final predicted category by selecting the class with the highest probability score. As illustrated in Figure 1, the system supports multiple categories such as Nature, Vehicles, People, Food, and Documents. This modular classification approach allows easy extension of categories in future versions of the system.

F. Automated Folder-Based Image Storage

Based on the predicted output shown in Figure 1, the image is automatically stored in the corresponding category folder (Nature Folder, Vehicles Folder, People Folder, Food Folder, or Documents Folder). This automated storage mechanism eliminates the need for manual sorting and enables efficient image organization. If the required folder does not exist, it is dynamically created by the system.

```
def store_image(img_path, predicted_class):
    dest_dir = os.path.join("classified_images",
                            predicted_class)
    os.makedirs(dest_dir, exist_ok=True)
    shutil.copy(img_path, dest_dir)
```

G. Output Display and System Termination

As indicated in the final stages of Figure 1, the system displays the predicted category along with its confidence score to the user. After successful classification and storage, the process terminates. This ensures both transparency of results and efficient organization of images.

IV. Result and Analysis

A. Results

The proposed mobile-based image classification system was evaluated using real-world images captured through the camera and gallery options of the Android application. The system correctly classified images into five categories—Vehicle, Nature, People, Food, and Document—based on their visual features. For each image, the predicted label and confidence score were displayed, and the image was automatically stored in the corresponding category folder. The included screenshots demonstrate representative outputs for each class, confirming accurate classification and effective automatic image organization.

B. Analysis

This section analyzes the learning behavior, performance trends, and generalization capability of the proposed mobile-based image classification model. The analysis is based on the training and validation metrics obtained during model training, as illustrated in Figure 7.

Fig. 2. Vehicle.

Fig. 3. Nature.

Fig. 4. Food.

Fig. 5. Document.

1) Model Training Results: The image classification model was trained using 20,668 training images and 3,255 validation images, distributed across five categories. Training was performed for five epochs, during which steady improvements in accuracy and a reduction in loss were observed.

As shown in Figure 7, the model rapidly learned meaningful visual features during the initial epochs, indicated by a sharp

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 12 | Dec - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

Fig. 6. People.

increase in training accuracy from Epoch 1 to Epoch 2. In later epochs, the learning curves gradually stabilized, demonstrating effective convergence. The use of transfer learning with MobileNetV2 enabled the model to achieve high performance within a limited number of epochs, significantly reducing training time while maintaining accuracy.

2) Training and Validation Performance: The training logs indicate consistent growth in both training and validation accuracy across epochs. The detailed performance is summarized as follows:

- **Epoch 1:** Training Accuracy = 71.7%, Validation Accuracy = 93.2%
- **Epoch 2:** Training Accuracy = 92.2%, Validation Accuracy = 95.7%
- **Epoch 3:** Training Accuracy = 94.4%, Validation Accuracy = 96.5%
- **Epoch 4:** Training Accuracy = 95.1%, Validation Accuracy = 97.0%
- **Epoch 5:** Training Accuracy = 95.6%, Validation Accuracy = 97.8%

As shown in Figure 7, the validation accuracy remains slightly higher than the training accuracy across most epochs. This behavior indicates strong generalization capability and confirms the absence of major overfitting. The higher validation accuracy can be attributed to the effective use of data augmentation techniques, which exposed the model to diverse image variations during training and improved robustness.

3) Generalization and Stability Analysis: The close alignment between the training and validation accuracy curves in Figure 7 suggests stable learning behavior. The absence of

Fig. 7. This figure shows training and validation accuracy.

large fluctuations or divergence between the curves confirms that the model is neither underfitting nor overfitting. The smooth and gradual improvement across epochs demonstrates that the chosen learning rate, optimizer, and training configuration are appropriate for the dataset.

4) Observations from Accuracy Trends: The rapid improvement in early epochs highlights the effectiveness of pre-trained features in MobileNetV2, while the gradual saturation in later epochs indicates that the model has reached an optimal learning state. Such behavior is desirable for mobile applications, as it balances accuracy with computational efficiency.

5) Summary of Analysis: Overall, the analysis confirms that the proposed model achieves efficient learning, strong generalization, and stable convergence. The training and validation accuracy trends shown in Figure 7 validate the effectiveness of transfer learning and data augmentation. These characteristics make the model well-suited for real-world mobile-based image classification and deployment within an Android application.

V. Conclusion and Future Scope

This paper presented a mobile-based image classification system capable of automatically categorizing images into multiple semantic classes, including Vehicle, Nature, People, Food, and Document. By leveraging transfer learning with the MobileNetV2 architecture, the proposed system achieved high classification performance while maintaining computational efficiency suitable for mobile devices. The training and validation results demonstrated stable learning behavior, strong generalization, and the absence of significant overfitting. The integration of the trained model into an Android application enabled real-time image classification and automatic organization of images into category-specific folders. Overall, the experimental results confirm the effectiveness and practicality of the proposed system for real-world mobile image management applications.

A. Future Scope

Although the proposed system demonstrates promising results, several enhancements can be explored in future work.

The number of image categories can be expanded to include additional real-world classes for broader applicability. Incorporating incremental or online learning techniques would allow the model to adapt to new data over time without complete retraining. Performance can be further improved by integrating advanced lightweight architectures or model optimization techniques such as quantization and pruning. Additionally, extending the system to support cloud-assisted classification, multilingual document recognition, and cross-platform deployment can enhance scalability and usability in diverse application scenarios.

REFERENCES

[1] S. R. Fanello, N. Noceti, G. Metta, and F. Odone, "Multi-Class Image Classification: Sparsity Does It Better," in *Proc. 8th Int. Conf. Comput. Vis. Theory Appl. (VISAPP)*, vol. 1, 2013, pp. 800-07.

[2] W. A. Ezat, M. M. Dessouky, and N. A. Ismail, "Multi-class Image Classification Using Deep Learning Algorithm," *J. Phys.: Conf. Ser.*, vol. 1447, no. 1, p. 012021, 2020.

[3] A. G. Howard et al., "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2017, pp. 4510-20.

[4] J. Zhang, Q. Zhang, J. Ren, Y. Zhao, and J. Liu, "Spatial-context-aware deep neural network for multi-class image classification," *arXiv preprint arXiv:2111.12296*, 2021.

[5] P. Murugan, "Implementation of Deep Convolutional Neural Network in Multi-class Categorical Image Classification," *arXiv preprint arXiv:1801.01397*, 2018.

[6] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *Adv. Neural Inf. Process. Syst. 25 (NIPS)*, 2012, pp. 1106-14.

[7] M. Tan and Q. V. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2019.

[8] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 770-78.