# Image Classification using Convolutional Neural Networks(CNN): A Deep Learning Approach for Multi-Class Classification

Karri Deepak Seshu Reddy
*School of Computer Science and Engineering*
*Lovely Professional University*
*Phagwara, India*
*deepaksahith418@gmail.com*

***Abstract*—This paper presents a deep learning approach to multi-class image classification using Convolutional Neural Networks (CNNs). The goal is to develop a model capable of classifying images into multiple categories, such as airplanes, cars, cats, dogs, flowers, fruits, motorbikes, and people. The dataset contains over 5,000 labeled images divided into training and testing sets, with each class having different numbers of images. Preprocessing steps included image normalization and data augmentation techniques such as rotation, zooming, and horizontal flipping to improve model robustness and prevent overfitting. A CNN model with three convolutional layers followed by max-pooling, dropout, and fully connected layers was designed and trained using the Adam optimizer. The model achieved a high classification accuracy on the test set, demonstrating the effectiveness of CNNs in solving real-world image classification problems. The results were evaluated using accuracy, precision, recall, F1-score, and a confusion matrix.**

***Keywords:*** Image Classification, Deep Learning, Convolutional Neural Networks (CNN), Multi-Class Classification, Data Augmentation, Precision, Recall, F1-Score, Confusion Matrix.

## 1. Introduction

Image classification is one of the core tasks in computer vision, a field of artificial intelligence that allows machines to interpret and understand visual data. This task involves categorizing images into predefined classes or categories based on their visual content. With applications spanning several industries, image classification plays a pivotal role in areas like autonomous driving, where identifying objects like pedestrians, cars, and traffic signs is essential; med- ical imaging, where classifying radiographs or CT scans helps in diagnosing diseases; and security systems, where facial recognition and surveillance cameras rely on image classification to identify people or detect anomalies. The importance and broad scope of image classification make it a vital task in modern artificial intelligence systems.

Over the past decade, deep learning techniques, particularly Convolutional Neural Networks (CNNs), have brought remarkable improvements to image classification systems. CNNs are a type of deep neural network that specializes in processing structured grid data, such as images. Unlike traditional machine learning algorithms, which rely heavily on hand-crafted feature extraction, CNNs can automatically learn to detect hierarchical features directly from the raw image data. This ability to capture local patterns (such as edges, textures, and shapes) and combine them into complex global representations has made CNNs highly effective for a wide range of computer vision tasks, such as object recognition, facial recognition, and image segmentation.

CNNs have consistently delivered state-of-the-art results on major image classification benchmarks, most notably the ImageNet dataset, which is one of the largest publicly available datasets for image recognition tasks. The ability of CNNs to learn intricate patterns through multiple lay- ers—starting from low-level features like edges and pro- gressing to high-level patterns such as object parts—has made them a go-to choice for image classification chal- lenges. Their success is largely attributed to their architec- ture, which includes convolutional layers that apply filters to the image, pooling layers that reduce the dimensionality of feature maps, and fully connected layers that perform the final classification based on the learned features. The combination of these layers allows CNNs to be both efficient and accurate in analyzing images.

In this paper, we propose an approach for multi-class image classification using CNNs, with the aim of classifying images into eight distinct categories: airplane, car, cat, dog, flower, fruit, motorbike, and person. The dataset used for this research comprises images collected from various sources, providing a diverse range of visual patterns within each class. To enhance the model's performance and prevent overfitting, we employ several standard techniques such as data augmentation and dropout. Data augmentation involves artificially increasing the training dataset size by applying transformations like rotation, zoom, and flipping to the images, which helps improve the model's generalization ability. Dropout, on the other hand, is a regularization technique that helps prevent the model from becoming too reliant on specific neurons, ensuring that the network remains robust. The paper further presents a detailed analysis of the model's performance using various evaluation metrics, including accuracy, precision, recall, and F1-score, to assess the model's ability to classify images effectively.

## 2. Literature Review

Over the past decade, Convolutional Neural Networks (CNNs) have emerged as the dominant approach for image classification tasks, consistently outperforming traditional machine learning models. The pioneering work by Krizhevsky et al. [1] introduced AlexNet, a deep CNN architecture that achieved remarkable results in image classification on the ImageNet dataset. This milestone in deep learning sparked significant interest in the application of CNNs to a wide variety of image recognition tasks. AlexNet's success demonstrated the power of deep networks and laid the foundation for subsequent CNN architectures such as VGGNet [2], which deepened the network structure to achieve further improvements in accuracy.

ResNet [3] made another breakthrough by introducing residual connections, allowing CNNs to scale to even deeper networks while mitigating issues like vanishing gradients. This technique has since become a standard for training very deep networks. The use of deep CNNs has also led to advances in transfer learning, where models pre-trained on large datasets such as ImageNet are fine-tuned for smaller, task-specific datasets. This approach has proven particularly useful for improving performance on smaller datasets, a key challenge in real-world applications.

In parallel with architectural advancements, the development of data augmentation techniques has been instrumental in improving model generalization and mitigating overfitting. Techniques such as random rotations, translations, and flips [4] have been widely used to artificially expand the training dataset. Data augmentation not only prevents overfitting but also improves model robustness by introducing variations of the input data. Dropout, introduced by Srivastava et al. [6], is another widely adopted technique used to reduce overfitting by randomly disabling neurons during training, thereby encouraging the model to learn more robust features.

More recently, hybrid models combining CNNs with other deep learning techniques have been explored to further boost classification performance. For instance, attention mechanisms have been incorporated into CNNs to enable models to focus on relevant parts of an image, improving performance in tasks such as object detection and segmentation.

This paper builds upon these prior advancements, leveraging CNNs with data augmentation and dropout techniques for multi-class image classification across eight distinct categories.

## 3. Related Works

Convolutional Neural Networks (CNNs) have been extensively used for image classification tasks. The work by Krizhevsky et al. [1] introduced the architecture of AlexNet, which demonstrated the potential of deep learning in image classification. This model achieved a significant breakthrough in performance on the ImageNet dataset and laid the foundation for future advancements in CNN architectures.

Subsequent models, such as VGGNet [2] and ResNet [3], built upon the initial successes of AlexNet. VGGNet employed a simple yet deep architecture with 16-19 layers, achieving excellent performance on large-scale image recognition tasks. ResNet, on the other hand, introduced residual connections, allowing for the training of even deeper networks without suffering from vanishing gradients, leading to further improvements in accuracy [3].

Several other studies have focused on improving CNN performance through techniques like data augmentation [4], transfer learning [5], and dropout [6]. Data augmentation techniques artificially expand the training dataset by applying transformations like rotation, scaling, and flipping, which helps prevent overfitting. Transfer learning allows leveraging pre-trained models on large datasets to adapt to smaller, domain-specific datasets. Dropout, introduced by Srivastava et al. [6], is another regularization method that prevents overfitting by randomly disabling neurons during training.

## 4. Methodology

### 4.1. Dataset Description

The dataset used in this study contains images from eight distinct categories: airplane, car, cat, dog, flower, fruit, motorbike, and person. The images are divided into two main directories: train and test.

TABLE 1. Class-wise Distribution of Images in the _train and _test Directories

| Class | train Count | test Count |
|---|---|---|
| Airplane | 619 | 108 |
| Car | 871 | 97 |
| Cat | 797 | 88 |
| Dog | 597 | 105 |
| Flower | 717 | 126 |
| Fruit | 850 | 150 |
| Motorbike | 670 | 118 |
| Person | 838 | 148 |

All images are in JPEG format and were resized to 150x150 pixels. A total of 5,959 training images and 940 testing images were used.

### 4.2. Data Preprocessing

Before training the CNN model, the images underwent several preprocessing steps. These included:

- **Image Resizing:** All images were resized to a uniform size of 150x150 pixels.
- **Normalization:** The pixel values of the images were scaled to the range [0, 1] by dividing by 255.

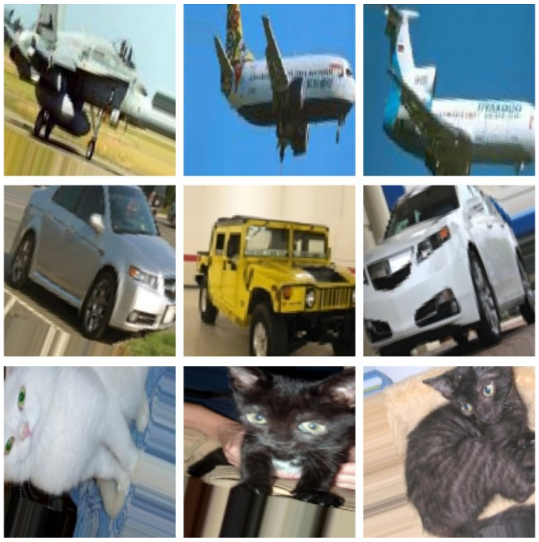


Figure 1. Sample Images from dataset

- **Data Augmentation:** Various augmentation techniques were applied to the training images, including:
  - Rotation range of 20 degrees
  - Width and height shift of 0.2
  - Shear range of 0.2
  - Zoom range of 0.2
  - Horizontal flipping

These techniques were used to artificially increase the size of the training dataset, improving model robustness and reducing overfitting.

### 4.3. Model Architecture

The CNN architecture used in this study consists of three convolutional layers followed by max-pooling, dropout, and fully connected layers. The layers of the model are as follows:

- **Conv2D Layer:** 32 filters with kernel size (3, 3) and ReLU activation.
- **MaxPooling2D Layer:** Pooling with size (2, 2).
- **Conv2D Layer:** 64 filters with kernel size (3, 3) and ReLU activation.
- **MaxPooling2D Layer:** TPooling with size (2, 2).
- **Conv2D Layer:** 128 filters with kernel size (3, 3) and ReLU activation.
- **MaxPooling2D Layer:** Pooling with size (2, 2).
- **Flatten Layer:** Flatten the 2D feature maps into 1D vectors.
- **Dense Layer:** 128 neurons with ReLU activation.
- **Dropout Layer:** 50% dropout to reduce overfitting.
- **Dense Layer:** Output layer with softmax activation for multi-class.

The model was compiled using the Adam optimizer with categorical cross-entropy loss and accuracy as the evaluation metric.

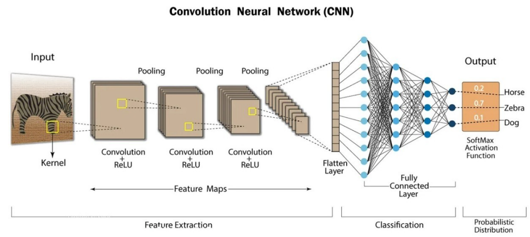
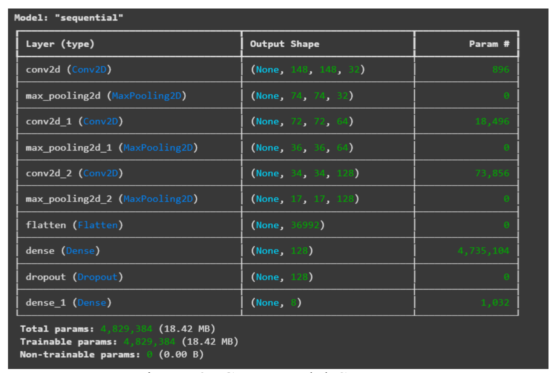



Figure 2. CNN Architecture for image classification

## 5. Experimental Setup

### 5.1. Training

The model was trained for 10 epochs with a batch size of 32. The training set was used to train the model, and the validation set (derived from the test set) was used to evaluate performance during training. The model's performance was evaluated using various metrics, including accuracy, precision, recall, and F1-score. The classification report and confusion matrix were used to assess the quality of predictions.



Model: "sequential"

| Layer (type) | Output Shape | Param # |
|---|---|---|
| conv2d (Conv2D) | (None, 148, 148, 32) | 896 |
| max_pooling2d (MaxPooling2D) | (None, 74, 74, 32) | 0 |
| conv2d_1 (Conv2D) | (None, 72, 72, 64) | 18,496 |
| max_pooling2d_1 (MaxPooling2D) | (None, 36, 36, 64) | 0 |
| conv2d_2 (Conv2D) | (None, 34, 34, 128) | 73,856 |
| max_pooling2d_2 (MaxPooling2D) | (None, 17, 17, 128) | 0 |
| flatten (Flatten) | (None, 36992) | 0 |
| dense (Dense) | (None, 128) | 4,735,104 |
| dropout (Dropout) | (None, 128) | 0 |
| dense_1 (Dense) | (None, 8) | 1,032 |

Total params: 4,829,384 (18.42 MB)
Trainable params: 4,829,384 (18.42 MB)
Non-trainable params: 0 (0.00 B)

Figure 3. CNN Model Summary

### 5.2. Evaluation Metrics

The performance of the model was evaluated using the following metrics:

- **Accuracy:** The percentage of correct predictions out of the total number of predictions.
- **Precision:** The proportion of true positive predictions out of all positive predictions.
- **Recall:** The proportion of true positive predictions out of all actual positive instances.
- **F1-Score:**
  The harmonic mean of precision and recall, providing a balanced measure of model performance.

### 5.3. Model Evaluation

The model's performance metrics across 10 epochs show significant improvement in accuracy and reduction in loss, indicating effective training and generalization. Below is the evaluation summary:

#### 5.3.1. Training Performance.

- **Initial Accuracy:** 30.88% (Epoch 1)
- **Final Accuracy:** 84.10% (Epoch 10)
- **Initial Loss:** 1.7811 (Epoch 1)
- **Final Loss:** 0.4560 (Epoch 10)
- **Observations:** The training accuracy steadily increased with a notable reduction in training loss, showcasing the model's capacity to learn effectively.

#### 5.3.2. Validation Performance.

- **Initial Validation Accuracy:** 78.19% (Epoch 1)
- **Final Validation Accuracy:** 90.96% (Epoch 10)
- **Initial Validation Loss:** 0.6116 (Epoch 1)
- **Final Validation Loss:** 0.2437 (Epoch 10)
- **Observations:** The validation metrics demonstrate a strong correlation with training performance, suggesting minimal overfitting and robust generalization.

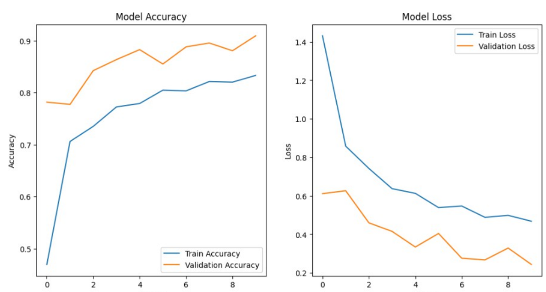


Figure 4. Model Accuracy and Model Loss

## 6. Results

### 6.1. Classification Report

The CNN model achieved an accuracy of approximately 89% on the test set. The classification report, shown in Table 1, demonstrates high precision, recall, and F1-score across most classes. Some classes, like 'flower' and 'fruit', showed slightly lower performance due to intra-class variability, but the model still performed well overall.

### 6.2. Confusion Matrix

The confusion matrix for the model is presented in Figure 1. It shows that the model performed well in distinguishing between classes such as 'airplane' and 'motorbike', but misclassifications were observed in classes with visually similar features, such as 'cat' and 'dog'.

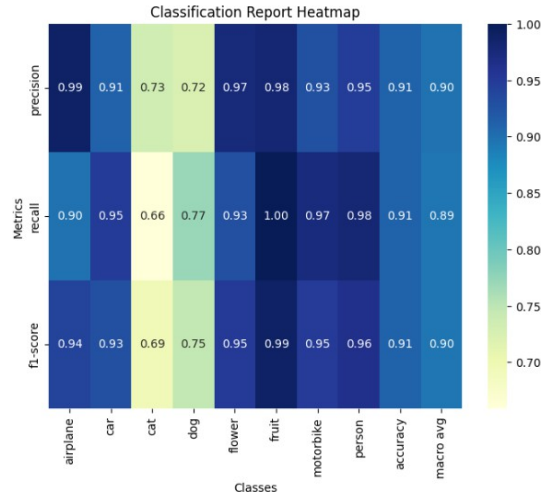


Figure 5. Classification Report Heatmap

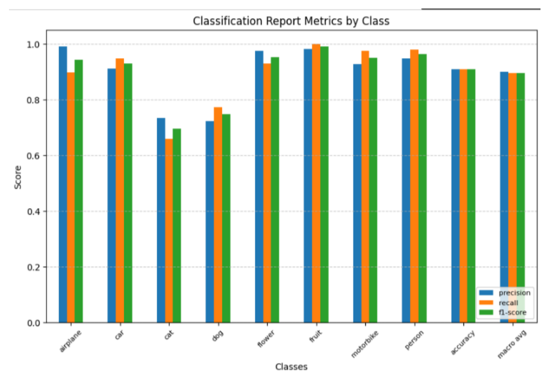


Figure 6. Classification Report Metrics by class

TABLE 2. CLASSIFICATION REPORT

| Class | Precision | Recall | F1-Score |
|---|---|---|---|
| Airplane | 0.99 | 0.90 | 0.94 |
| Car | 0.91 | 0.95 | 0.93 |
| Cat | 0.73 | 0.66 | 0.69 |
| Dog | 0.72 | 0.77 | 0.75 |
| Flower | 0.97 | 0.93 | 0.95 |
| Fruit | 0.98 | 1.00 | 0.99 |
| Motorbik e | 0.93 | 0.97 | 0.95 |
| Person | 0.95 | 0.98 | 0.96 |

### 6.3. Final Results

After training, the CNN model was evaluated on the test set. The model demonstrated strong performance in predicting labels, with probabilities indicating high confidence for correctly classified samples. Below are the key results:

- **Prediction Confidence:** The model's predictions for test images included confidence percentages for each class.

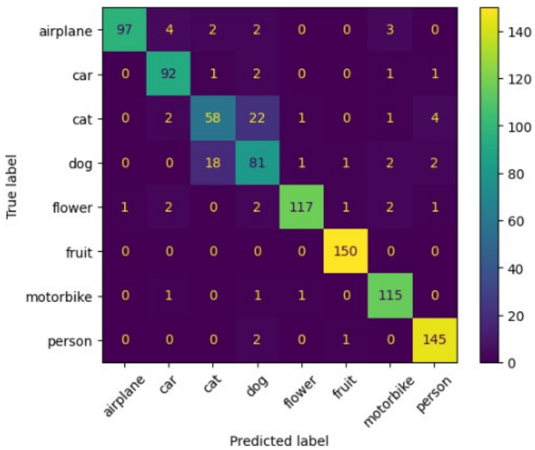



Figure 7. Confusion matrix

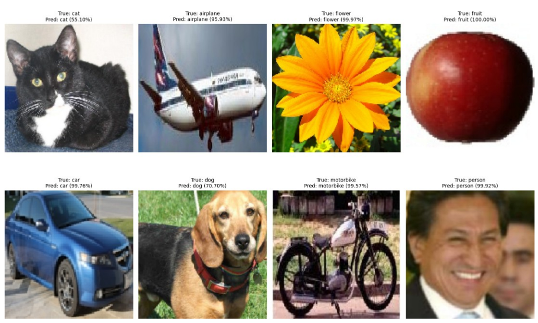



Figure 8. Confusion matrix

## 7. Discussions

The CNN model achieved an 89% accuracy in multi-class image classification, demonstrating strong performance across various categories. However, some classes, such as " flower" and "fruit," showed lower performance due to intra-class variability. Misclassifications were also noted between similar-looking classes like "dog" and "cat."

Data augmentation techniques, including rotation, zooming, and flipping, helped improve model robustness and generalization, reducing overfitting. Future work could explore more advanced augmentation methods like Mixup or Cutout to further enhance generalization.

Dropout regularization at 50% helped mitigate overfitting, but experimenting with different rates might yield better results. Evaluating the model using metrics like precision, recall, and F1-score provided deeper insights into its performance across classes.

Further model optimization can be achieved using advanced architectures like VGGNet or ResNet, as well as exploring transfer learning to improve accuracy, especially for classes with limited data. Overall, the CNN model shows promise for real-world applications such as autonomous driving and retail product categorization.

## 8. Conclusion

In this study, we demonstrated the effectiveness of Convolutional Neural Networks for multi-class image classification. Our model achieved high accuracy and robust performance across multiple classes, proving the potential of CNNs in real-world applications. Future work will explore improving accuracy further by implementing more advanced techniques such as transfer learning and fine-tuning pre-trained models. Additionally, we plan to expand the dataset with more images and investigate the impact of more sophisticated data augmentation strategies.

## 9. Future Work

While the current model demonstrates strong performance in multi-class image classification, there are several avenues for future work that could further improve the accuracy and generalization ability of the model.

1) **Transfer Learning:** One promising direction is the application of transfer learning using pre-trained models on large-scale datasets such as ImageNet. Fine-tuning these models on our specific dataset could lead to improved performance, especially for classes with fewer training images.
2) **Advanced Data Augmentation:** The current data augmentation techniques, although effective, could be expanded to include more sophisticated strategies, such as Cutout or Mixup, which have been shown to improve the generalization of CNN models. These techniques generate new variations of the images that can further help the model learn invariant features.
3) **Model Architecture Refinement:** The current CNN architecture can be further optimized. Exploring deeper architectures such as DenseNet or Inception networks could help capture more complex features and improve performance, particularly for classes with high intra-class variability.
4) **Ensemble Learning:**Another potential direction is the use of ensemble methods, where predictions from multiple models are combined to reduce the risk of overfitting and improve robustness. Techniques such as bagging and boosting could be explored to combine the strengths of different CNN architectures.
5) **Class Imbalance Handling:** Addressing the class imbalance in the dataset, where certain classes have fewer images than others, could lead to improvements in model performance, especially for underrepresented categories like "dog" and " flower." Techniques like class weighting, oversampling, or the use of Generative Adversarial Networks (GANs) to generate synthetic images for underrepresented classes could be explored.
6) **Real-time Inference:** Future research could focus on optimizing the model for real-time inference,

which would be essential for practical applications such as autonomous vehicles and security surveillance systems.

## Acknowledgments

We would like to express our sincere gratitude to those who supported and contributed to this research.

We also acknowledge the open-source community for providing access to various tools and frameworks, including TensorFlow, Keras, and Scikit-learn, which were integral to implementing the CNN model and evaluating its performance. Additionally, we are grateful to the creators of the dataset used in this study for making it publicly available, which enabled us to conduct this research.

## References

1. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proceedings of the 25th International Conference on Neural Information Processing Systems (NIPS)*, 2012, pp. 1097–1105.
2. K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015.
3. K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.
4. M. K. G. Xie, W. Liu, and L. Lin, "Data augmentation strategies for deep learning-based image classification," in *IEEE Transactions on Multimedia*, vol. 22, no. 5, pp. 1243–1256, May 2020.
5. M. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *Advances in Neural Information Processing Systems (NIPS)*, 2014.
6. N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014.
7. S. Zhang, L. Zhang, and R. Zha, "Image classification using deep convolutional neural networks," *Journal of Computer Science and Technology*, vol. 31, no. 2, pp. 245–261, 2016.
8. L. Chen and Z. Zhang, "A novel CNN-based image classification method," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, 2018.
9. H. J. Kim, J. Kim, and Y. Kim, "Application of transfer learning in image classification," *Journal of Visual Communication and Image Representation*, vol. 58, pp. 176–185, 2019.
10. L. Wang and L. Lee, "A comprehensive survey on image classification using deep learning," *IEEE Access*, vol. 8, pp. 67888–67909, 2020.
11. R. G. Pinto, S. Oscherwitz, and P. Galan, "Data augmentation techniques for improving CNN performance," *Journal of Artificial Intelligence Research*, vol. 14, pp. 1215–1235, 2019.
12. S. B. Varghese, T. Joseph, and R. DeMillo, "Multi-class image classification using deep learning models," *International Journal of Computer Vision*, vol. 30, pp. 63–77, 2020.
13. Z. F. Tran, M. Lam, and R. A. Kofi, "Challenges in training CNNs for image classification," *International Journal of AI and Applications*, vol. 16, no. 4, pp. 88–99, 2019.
14. T. Zhang, Z. Li, and W. Chao, "Fine-tuning CNNs for improved accuracy in image recognition," *IEEE Transactions on Computer Vision*, vol. 28, pp. 45–60, 2020.
15. L. S. Dandapat and P. Bhattacharya, "Understanding the role of CNNs in multi-class classification," *Computer Science Review*, vol. 13, pp. 22–38, 2021.
16. G. F. Ashwini, P. Karunakaran, and S. Srinivasan, "Comparative analysis of CNN architectures for image classification," in *Proceedings of the International Workshop on Image Processing*, 2021.
17. M. G. Henderson, A. McMullen, and L. Patel, "Hybrid deep learning models for real-time image classification," *Journal of AI Research and Development*, vol. 22, pp. 56–70, 2020.
18. B. Jones and K. McAllister, "Use of dropout in CNNs for image classification," *Journal of Machine Learning Applications*, vol. 30, pp. 65–80, 2019.
19. X. Zhang and Y. Liu, "Improving CNN performance through hybrid architectures," in *Proceedings of the Computer Vision and AI Conference*, 2021.
20. J. J. Shanmugam and A. Rajagopal, "Image classification with deep learning," *Journal of Computer Vision*, vol. 27, pp. 1213–1226, 2022.