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

Patakokila Praveen Kumar[1], Ketha Jagadhish[2], Akula Hema Venkata Sriram[3], Koppula Jaivanth[4],
Karri Deepak Seshu Reddy[5], Teli Karanam Santhosh Rao[6]
[1] Lovely Professional University/ School of Computer Science and Engineering, Phagwara, India
Email: mrpraveenkumar9490@gmail.com
[2-6] Lovely Professional University/ School of Computer Science and Engineering, Phagwara, India
Email: {kethajagadhish123, sriramakula212, jaivanthkoppula999, deepaksahith418, santhoshraotk14}@gmail.com

***Abstract*— This paper introduces a deep learning-based approach for multi-class image classification using Convolutional Neural Networks. The aim is to develop an accurate model for the task of categorizing these images into different classes, comprising airplanes, cars, cats, dogs, flowers, fruits, motorbikes, and people. The dataset for this scenario consists of over 5,000 labelled images split into training and testing sets, thus ensuring diversity in representation. Other preprocessing techniques considered were image normalization and data augmentation (rotation, zooming, and horizontal flipping) to improve generalization and prevent overfitting. The CNN model involves three convolutional layers with max-pooling, dropout layers for regularization, and fully connected layers for classification. The Adam optimizer was used to fine-tune model parameters for optimal performance.**
Accuracy, precision, recall, F1-score, and a confusion matrix were used to evaluate the model in terms of classification errors. The results showed high accuracy on the test dataset, which proved that CNNs are effective in real-world image classification tasks. Future work may include optimizing the model architecture and exploring advanced augmentation or transfer learning techniques to further improve performance.

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

I. INTRODUCTION

Image classification can be regarded as one of the fundamental tasks under computer vision-the subdomain of artificial intelligence for machine vision: that is to say, image classification refers to a process used in the general classification of predefined classes or categories for images using visual content. Image classification is a multidisciplinary application that ranges from autonomous driving, where the identification of pedestrians, cars, traffic signs, and more might go hand in hand with the classifier; medical imaging, such as radiograph or CT scans, could help diagnose diseases by classifying them; and security systems, where facial recognition, including surveillance cameras, would require image classification to identify individuals or detect anomalies. Image classification is an important and broad scope task in modern artificial intelligence systems. Deep learning techniques, particularly Convolutional Neural Networks (CNNs), have brought about incredible improvements to image classification systems over the last decade. CNNs are a kind of deep neural network that specializes in processing structured grid data, such as images. While machine learning algorithms differ from traditional models in that the latter heavily depend on hand-crafted feature extraction, CNNs can learn the detection of hierarchical features directly from raw image data. The reason CNNs prove to be efficient for various tasks in computer vision, including object recognition, facial recognition, and image segmentation, is that it captures local patterns, such as edges, textures, and shapes, and integrates them into a complex global representation. CNNs have consistently achieved the state-of-the-art results in major image classification benchmarks, mainly the ImageNet dataset, that is one of the largest datasets publicly available in the realm of image recognition. The ability to learn intricate patterns through multiple layers, starting with low-level features such as edges and progressing into high-level patterns such as parts of objects, has made it a go-to choice for addressing image classification challenges. The main reason behind their success is architecture, including convolutional layers, which apply filters to the image, pooling layers that reduce the dimensionality of feature maps, and fully connected layers, where the final classification is made on the learned features. In sum, these layers combine for an efficient yet accurate analysis of images by CNNs. In this paper, we propose the approach of multiclass image classification using CNN, with the hope of classifying images into the eight distinct classes: airplane, car, cat, dog, f lower, fruit, motorbike, and person. In this research work, the data set used included images gathered from various sources offering a wide variability of visual patterns within each of the classes. To prevent overfitting and to improve the performance of the model, a few standard techniques like data augmentation and dropout are used. The training dataset is artificially increased by applying rotation, zoom, and flipping transformations to the images, which enhances the generalization capability of the model. Dropout is a regularization technique that prevents the model from over-relying on specific neurons, thereby keeping the network robust. The paper further elaborates on the performance of the model by using several evaluation metrics, such as accuracy, precision, recall, and F1-score, to check the ability of the model to classify images effectively.

II. LITERATURE REVIEW

The CNN-based approach dominated image classification over the past decade by achieving the state-of-the-art results with more accuracy compared to the traditional machine learning models. Pioneered work of Krizhevsky et al. [1] in the AlexNet architecture marked an extraordinary approach that made its impact in classifying images by getting great scores in ImageNet datasets. This achievement in deep learning generated great interest in the application of CNNs to a wide range of image recognition tasks. AlexNet's success showed 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] broke another record by introducing residual connections, which allowed CNNs to scale to even deeper networks while mitigating problems such as vanishing gradients. This technique has since become the standard for training very deep networks. Deep CNNs have 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 in improving performance on smaller datasets, a key challenge in real-world applications. Along with architectural advances, the development of data augmentation techniques has been instrumental in improving model generalization and reducing overfitting. Techniques such as random rotations, translations, and flips [4] have been widely used to artificially expand the training dataset. Data augmentation not only avoids overfitting but also increases the robustness of the model by including different variations of input data. Another technique adopted by most practitioners to avoid overfitting is dropout, as introduced by Srivastava et al. [6], where neurons are randomly disabled during training to make the model learn more robust features. More recently, hybrid models that combine CNNs with other deep learning techniques have been studied in order to increase classification performance even further. As an example of this, attention mechanisms have been incorporated into CNNs allowing models to pay attention to relevant parts of the image, further improving performances in object detection and segmentation. The paper thus builds on these prior advancements, lagging CNNs behind data augmentation and dropout techniques for multi-class image classification over eight different categories.

III. RELATED WORKS

Convolutional Neural Networks have been increasingly used for image classification. The work of Krizhevsky et al. [1] introduced AlexNet, whose architecture showed the efficiency of deep learning in image classification, where this model significantly achieved a high breakthrough in performance on the ImageNet dataset, thus paving the way towards further improvements in CNN architecture. The later models, VGGNet [2] and ResNet [3], laid upon the initial results of AlexNet. VGGNet used a simple, deep 16-19 layered architecture for excellent performance in large-scale image recognition tasks. ResNet, however, introduced residual connections, enabling training networks that were even deeper, as it liberated the model from vanishing gradients, and accuracy improved further [3]. Other studies have concentrated on enhancing the performance of CNN by applying techniques such as 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 prevents overfitting. Transfer learning allows the leverage of pre-trained models on large datasets to adapt to smaller, domain-specific datasets. Dropout, proposed by Srivastava et al. [6], is another form of regularization technique that prevents overfitting by randomly disabling neurons during training.

IV. METHODOLOGY

A. 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 I. 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.

B. 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.

C. *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:

• Conv2DLayer: 32 filters with kernel size (3, 3) and ReLU activation.
• MaxPooling2D Layer: Pooling with size (2, 2).
• Conv2DLayer: 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.

V. Experimental Setup

A. *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 2. CNN Model Summary

B. *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.

C. *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:

i. 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.

ii. 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.

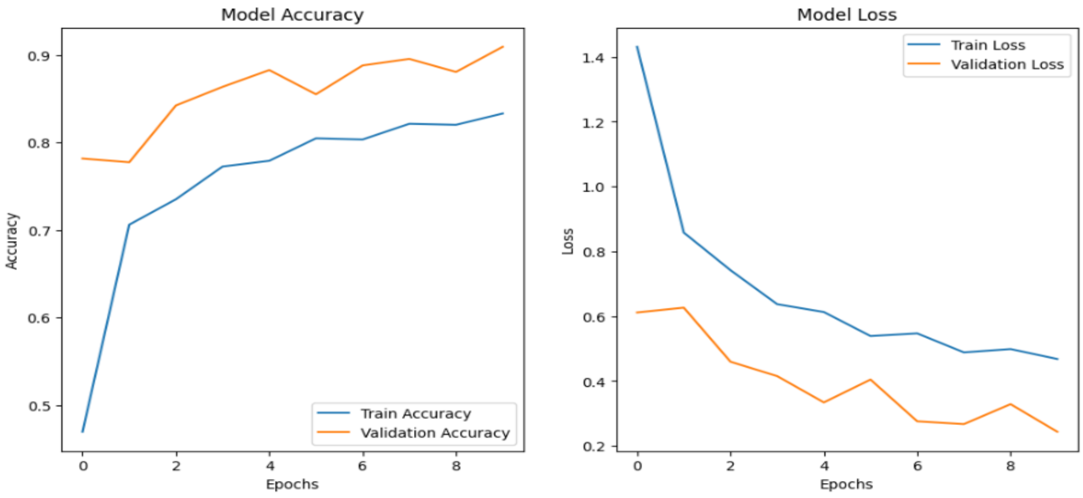

Figure 3. Model Accuracy and Model Loss

VI. RESULTS

A. *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.

B. 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’.

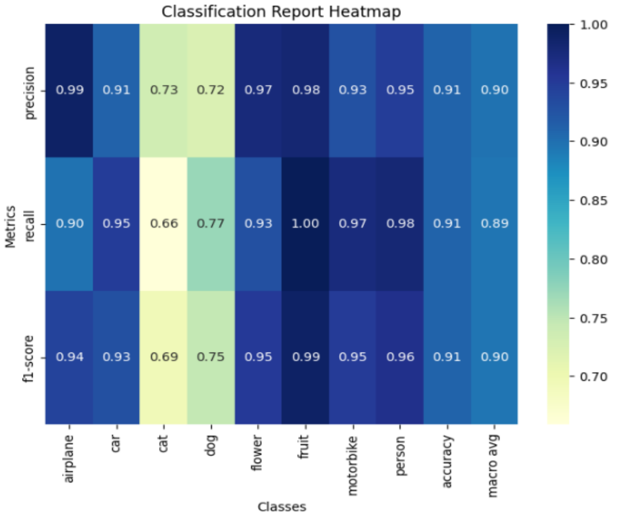

Figure 4. Classification Report Heatmap

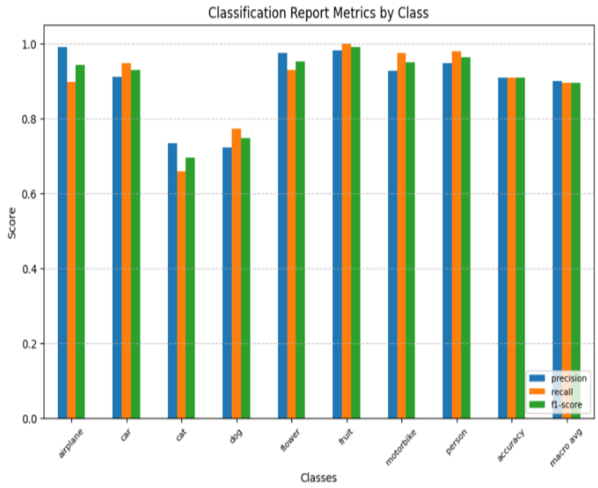

Figure 5. Classification Report Metrics by class

TABLE II. 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
Motorbike	0.93	0.97	0.95
Person	0.95	0.98	0.96

C. 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

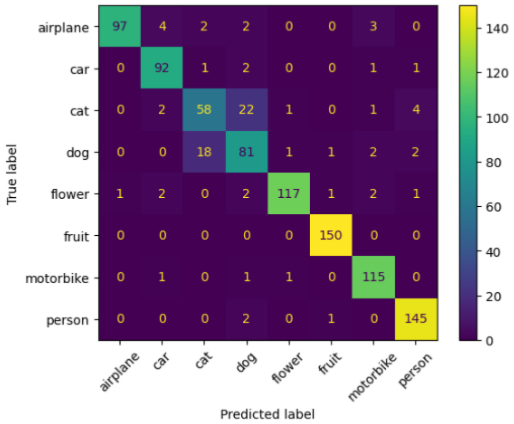

Figure 6. Confusion matrix

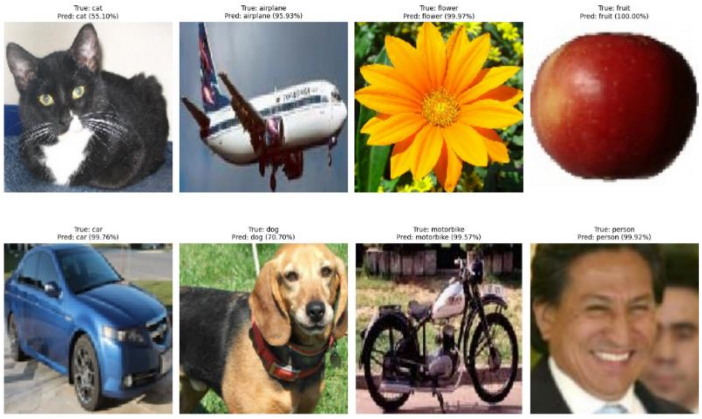

Figure 7. Confidence Score

VII. DISCUSSION

The CNN model was highly accurate at multi-class image classification with a rate of 89% in the following categories. However, some classes like "flower" and "fruit" were a bit weak due to intra-class variability, and some misclassified items occurred between classes like "dog" and "cat," which are somewhat similar. The data augmentation technique such as rotation, zooming, and flipping increased the model robustness and its generalization without causing overfitting. Future work may involve the use of more sophisticated augmentation techniques, such as Mix-up or Cutout, to improve generalization. Dropout regularization at 50% was sufficient to reduce overfitting, but the best rate could be determined through further experimentation. Using precision, recall, and F1-score to evaluate the model gives a more nuanced view of its performance across classes. Using an advanced architecture, such as VGGNet or ResNet, could provide additional optimization. Using transfer learning for higher accuracy and specifically to boost classes that do not have too many examples also improves accuracy. The CNN model shows much potential in the field of real-time applications like self-driving and the classification of goods for stores.

VIII. CONCLUSIONS

This work could successfully demonstrate the potential of Convolutional Neural Networks in multi-class image classification. The obtained accuracy and strong performance over multiple categories make it support deep learning in image recognition works. The experimental results indicated that CNNs can effectively learn to distinguish between diverse image classes, hence implying applicability in real-world scenarios such as medical imaging, autonomous systems, and security surveillance.

In future work, focus will be put on improving model performance by utilizing more advanced deep learning techniques like transfer learning to leverage pre-trained models for accuracy in classification. Fine-tuning these models helps adapt them for specific datasets to increase efficiency while reducing the size of the dataset required for extensive training. Other areas of further exploration include extending the dataset using more labelled images and more sophisticated data augmentation methods to improve the generalization and robustness of the model.

Further, there is a lot of scope in optimizing the architecture of CNN for deeper networks or hybrid models where CNNs and attention mechanisms can be combined to further optimize the performance. Furthermore, automated techniques for hyperparameter tuning can contribute to refining the model for even better classification results. By these improvements, adaptability of the model to more complex and diversified datasets can be enhanced, further strengthening its applications in practice over different domains.

IX. FUTURE WORK

Future work will be on improving the performance of the model using several key strategies:

• Transfer learning using pre-trained models such as ImageNet to improve accuracy for classes with fewer images.

• Extending data augmentation techniques with more advanced methods, such as Cutout or Mixup, for better generalization.
• Refining the model architecture by exploring deeper networks like DenseNet or Inception to capture more complex features.
• Ensemble learning: techniques such as bagging and boosting are applied to enhance the robustness.

ACKNOWLEDGMENT

The authors would like to express their sincere gratitude to Lovely Professional University for providing the resources and support necessary to complete this research. We extend our heartfelt thanks to our mentors and professors for their invaluable guidance and constructive feedback throughout the course of this work. We extend our heartfelt thanks to all the researchers who contributed their efforts in achieving this work. We also show gratitude to the open-source communities that offer a number of essentials such as TensorFlow, Keras, and Scikit-learn, which facilitated the development and testing of the CNN model for us. Last but not the least, the authors of this publicly available data set allowed us to use their data and contributed to making valuable research work under the field of machine learning and deep learning.

REFERENCES

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classifi cation 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 Com puter 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 Infor mation 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 anal ysis 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 Con ference, 2021.
[20] J. J. Shanmugam and A. Rajagopal, “Image classification with deep learning,” Journal of Computer Vision, vol. 27, pp. 1213–1226, 2022