IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁙ Peer-reviewed & Refereed journal ⁙ Vol. 15, Issue 4, April 2026

DOI: 10.17148/IJARCCE.2026.154285

DEVELOPMENT OF A CONVOLUTIONAL NEURAL NETWORK FOR BINARY IMAGE CLASSIFICATION OF BEARS AND PANDAS WITH APPLICATIONS IN DIGITAL IMAGE PROCESSING

P Aarti Pai[1], Khushi[2], Sudarshan SR[3], H Mary Shyni[4]

Student, Department of Computer Science and Engineering,

SRM Institute of Science and Technology, Vadapalani Campus, Vadapalani, Chennai, India[1-3]

Mentor, Department of Computer Science and Engineering,

SRM Institute of Science and Technology, Vadapalani Campus, Vadapalani, Chennai, India[4]

Abstract: The study investigates the design, application, and evaluation of a CNN model that is trained to perform binary classification based on bear and panda images. The use of an open-source dataset allows integrating certain basic ideas behind digital image processing together with deep learning techniques, which can be applied during image preprocessing and classification. The proposed CNN model comprises convolution and pooling layers followed by fully connected layers that can be used for binary classification. The Adam optimizer along with the binary cross entropy loss function were used, whereas accuracy for training the model, precision, and recall were used to measure the performance of the developed model.

Keywords: Image Processing, Binary Image Classification, Convulutional Neural Network

I. INTRODUCTION

Classification of Images is an important issue in Digital Image Processing and Computer Vision fields that are used in various applications such as ecology, surveillance, and automation. It is known that there can be issues regarding the accurate classification of images based on similarity between some images like bear and panda due to their similarities and environmental differences. CNN, Convolutional Neural Network, is a kind of Deep Learning method that works based on the way humans see and has caused a revolution in the field of image classification using automatic feature detection and hierarchy learning of features.

In this research, a CNN classifier will be constructed to classify binary bear-panda image dataset with the application of image processing operations. The data set utilized for this study will be obtained from Kaggle.

II. LITERATURE REVIEW

A. Image Processing Techniques and Deep Learning

There are many articles discussing how image processing techniques such as image resizing, normalization, and augmentation are used in pre-processing.[1][5][6] These techniques are significant in solving the issue of overfitting and detecting features. Nevertheless, some papers fail to examine in detail how each method influences the generalization capability of deep learning models. Hence, there should be more investigations in this regard.

B. Deep CNN Model for Species Image Recognition

Many different models have been applied in prior literature depending on the CNN depth. The models range from shallow neural network architectures to deep CNNs (e.g., ResNet, EfficientNet). [2] Despite achieving higher accuracy using deep CNNs, these models consume much computational power and may lead to overfitting when training data sets are minimal. [3][10]

C. Solving the Issue of Class Similarity and Complexity in Background

One of the challenges faced in identifying species is recognizing similar animals. Some scholars try to address the issue of class similarity by applying edge detection algorithms and texture analysis approaches. [4] Moreover, efforts have been made to convert the input data to another space color domain. [9] Yet very little research has been done in this area.

D. External Augmentation and Data Augmentation Usage

Data augmentation performed through external sources or artificial creation of images has proved to be very beneficial when trying to achieve better generalizability of the models.[8] However, it is very difficult to find out what kind of augmentation was done and how it affected the results during testing. [5]

E. Criteria For Evaluations

Accuracy, precision, recall, and confusion matrix values can be taken into account as criteria for evaluations [4] as they provide valuable information regarding the performance of the model but at times they do not include F1 score calculations or the balance between type I and type II errors, which is essential when it comes to applying the algorithm to monitor wildlife species. [9]

F. Possible Criticism

It is very important to state more explicitly how exactly the datasets were built, specifically paying attention to the augmentation process and external data usage. [6][8]

Conducting comparative studies on various Convolutional Neural Networks designed for wildlife image recognition could also be helpful.

III. METHODOLOGY

A. Dataset Description

The dataset used is publicly accessible on Kaggle (https://www.kaggle.com/datasets/mattop/panda-or-bear-image-classification) and includes labeled images of bears and pandas. These images have different lightings and backgrounds, similar to real-life scenarios. The dataset comprises two folders for each category (Refer to Fig. 1). Augmented images not included in the testing dataset were added externally for diverse testing purposes.

Fig. 1 Sample images from the dataset showing bears (left) and pandas (right).

B. Image Preprocessing

1) Image Resizing:

All images have been resized to 256 by 256 pixels to ensure uniformity in image size while training the CNN.

2) Image Normalization:

The pixel values have been normalized to the range [0,1] to ensure stable model training.

3) Image Data Augmentation:

Data augmentation techniques like flipping, rotation, and zooming are used to avoid overfitting by increasing the size of the data set.

4) *RGB Channel Maintenance:*
The red, green, and blue color channels have been preserved since they contain essential information for differentiating bears from pandas.

5) *Image Augmentation for Testing:*
The testing dataset was augmented with few publicly available real-life photos to help better evaluate the model.

C. *Model Architecture*

1) *Input Layer*
Input dimension varies according to backend conventions.

2) *Convolution Layers*
Consists of three Convolution Layers, filter sizes 16, 32, and 64 respectively. Each of these uses a kernel size (2,2) and are activated using the ReLU activation function.

3) *Max Pooling Layers*
Each convolution layer is followed by a pooling layer.

4) *Flatten Layer*
Converts 3D (Image Dimensions + Color Channels) Feature map into 1D vector.

5) *Fully Connected (Dense) Layer*
Contains 64 neurons activated by the ReLU activation function.

6) *Dropout Layer*
The Dropout Layer has a 50% dropout to prevent overfitting. [7]

7) *Output Layer*
Contains one output neuron activated by the sigmoid activation function to provide the result.

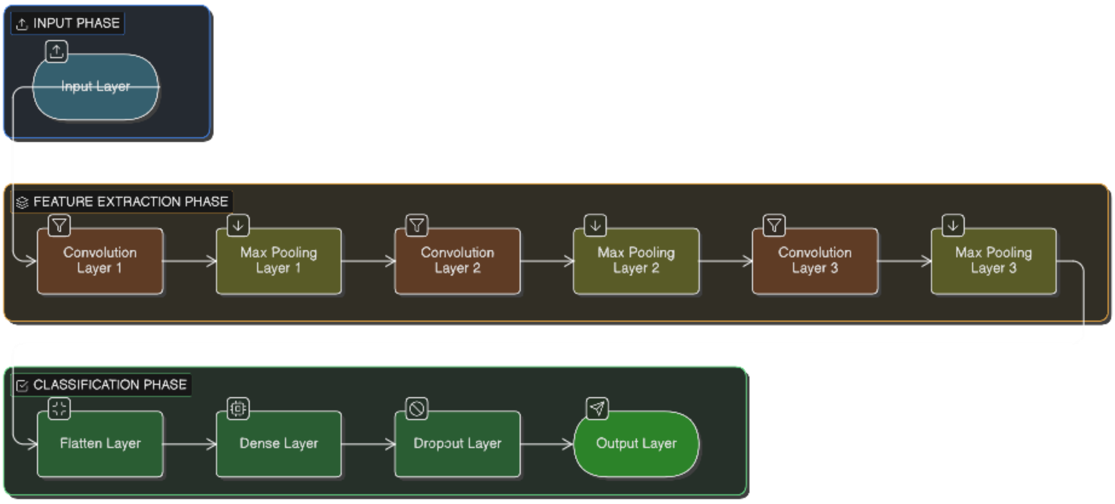


Fig. 2 Diagram showing the architecture of CNN with layers.

TABLE 1 TRAINING CONFIGURATION

Loss Function	Binary Cross-Entropy
Optimizer	Adam Optimizer with adaptive learning rate
No. Of Epochs	10
Batch Size/Epoch	16 images
Metrics Used	Accuracy, Precision, and Recall

D. *Implementation Environment*

The model was implemented in Python using the TensorFlow and Keras libraries. Data loading and preprocessing utilize Keras's image_dataset_from_directory for efficient pipeline management.

1) Digital Image Processing Integration

Digital image processing techniques are critical to the success of CNN models. In this study:

2) Image Rescaling and Normalization

These standardize inputs, reducing variance and improving convergence.

3) Augmentation

Artificially increases dataset size and diversity, simulating real-world image variations.

4) Edge and Texture Features

The small kernel size (2x2) in convolutional layers allows capturing fine-grained texture and edge details, essential for differentiating fur patterns and facial features between bears and pandas.

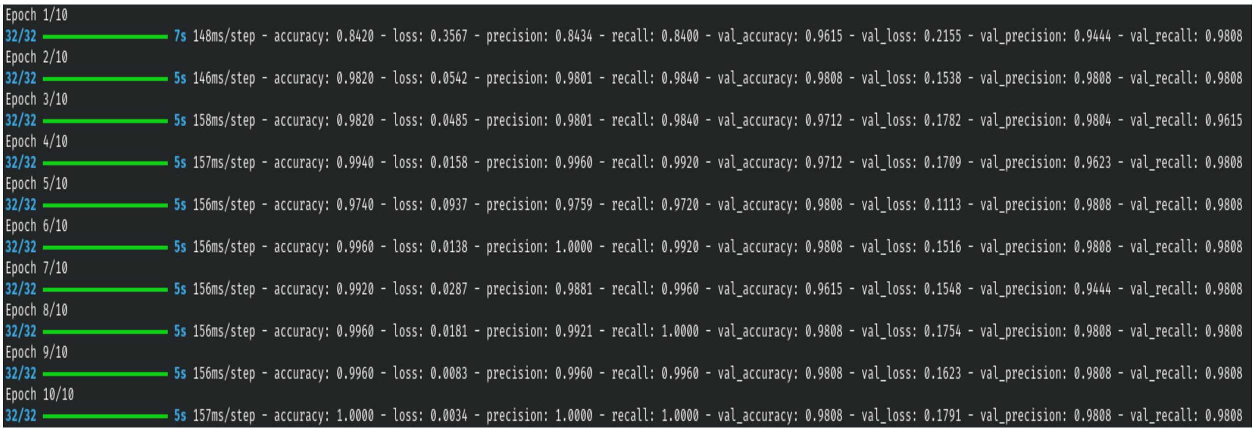
The interplay between preprocessing and CNN feature extraction exemplifies how digital image processing enhances deep learning efficacy.

IV. RESULTS

A. *Training Performance*

The model demonstrated steady improvement across epochs, achieving high accuracy, and balanced precision and recall.

Figures 3-6 illustrate the various statistics of the model during training and validation



```
Epoch 1/10
32/32 ━━━━━━━━━━━━━━━━━━━━ 7s 148ms/step - accuracy: 0.8420 - loss: 0.3567 - precision: 0.8434 - recall: 0.8400 - val_accuracy: 0.9615 - val_loss: 0.2155 - val_precision: 0.9444 - val_recall: 0.9808
Epoch 2/10
32/32 ━━━━━━━━━━━━━━━━━━━━ 5s 146ms/step - accuracy: 0.9820 - loss: 0.0542 - precision: 0.9801 - recall: 0.9840 - val_accuracy: 0.9808 - val_loss: 0.1538 - val_precision: 0.9808 - val_recall: 0.9808
Epoch 3/10
32/32 ━━━━━━━━━━━━━━━━━━━━ 5s 158ms/step - accuracy: 0.9820 - loss: 0.0485 - precision: 0.9801 - recall: 0.9840 - val_accuracy: 0.9712 - val_loss: 0.1782 - val_precision: 0.9804 - val_recall: 0.9615
Epoch 4/10
32/32 ━━━━━━━━━━━━━━━━━━━━ 5s 157ms/step - accuracy: 0.9940 - loss: 0.0158 - precision: 0.9960 - recall: 0.9920 - val_accuracy: 0.9712 - val_loss: 0.1709 - val_precision: 0.9623 - val_recall: 0.9808
Epoch 5/10
32/32 ━━━━━━━━━━━━━━━━━━━━ 5s 156ms/step - accuracy: 0.9740 - loss: 0.0937 - precision: 0.9759 - recall: 0.9720 - val_accuracy: 0.9808 - val_loss: 0.1113 - val_precision: 0.9808 - val_recall: 0.9808
Epoch 6/10
32/32 ━━━━━━━━━━━━━━━━━━━━ 5s 156ms/step - accuracy: 0.9960 - loss: 0.0138 - precision: 1.0000 - recall: 0.9920 - val_accuracy: 0.9808 - val_loss: 0.1516 - val_precision: 0.9808 - val_recall: 0.9808
Epoch 7/10
32/32 ━━━━━━━━━━━━━━━━━━━━ 5s 156ms/step - accuracy: 0.9920 - loss: 0.0287 - precision: 0.9881 - recall: 0.9960 - val_accuracy: 0.9615 - val_loss: 0.1548 - val_precision: 0.9444 - val_recall: 0.9808
Epoch 8/10
32/32 ━━━━━━━━━━━━━━━━━━━━ 5s 156ms/step - accuracy: 0.9960 - loss: 0.0181 - precision: 0.9921 - recall: 1.0000 - val_accuracy: 0.9808 - val_loss: 0.1754 - val_precision: 0.9808 - val_recall: 0.9808
Epoch 9/10
32/32 ━━━━━━━━━━━━━━━━━━━━ 5s 156ms/step - accuracy: 0.9960 - loss: 0.0083 - precision: 0.9960 - recall: 0.9960 - val_accuracy: 0.9808 - val_loss: 0.1623 - val_precision: 0.9808 - val_recall: 0.9808
Epoch 10/10
32/32 ━━━━━━━━━━━━━━━━━━━━ 5s 157ms/step - accuracy: 1.0000 - loss: 0.0034 - precision: 1.0000 - recall: 1.0000 - val_accuracy: 0.9808 - val_loss: 0.1791 - val_precision: 0.9808 - val_recall: 0.9808
```

Fig. 3 Training metrics over epochs.

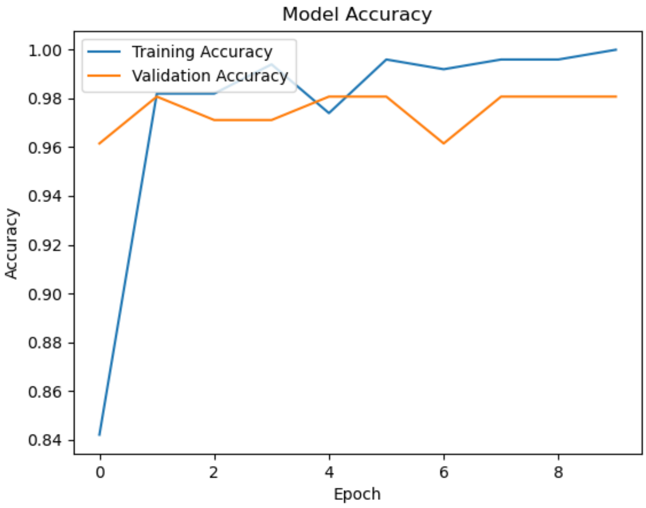


Fig. 4 Training and Validation Accuracy per Epoch.

DOI: 10.17148/IJARCCE.2026.154285

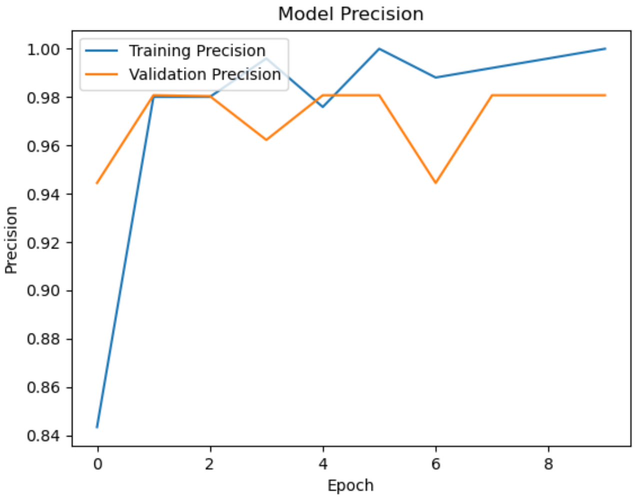


Fig. 5 Training and Validation Precision per Epoch.

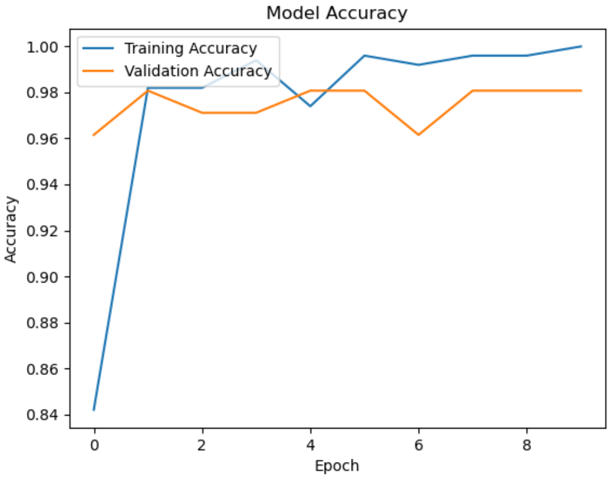


Fig. 6 Training and Validation Recall per Epoch.

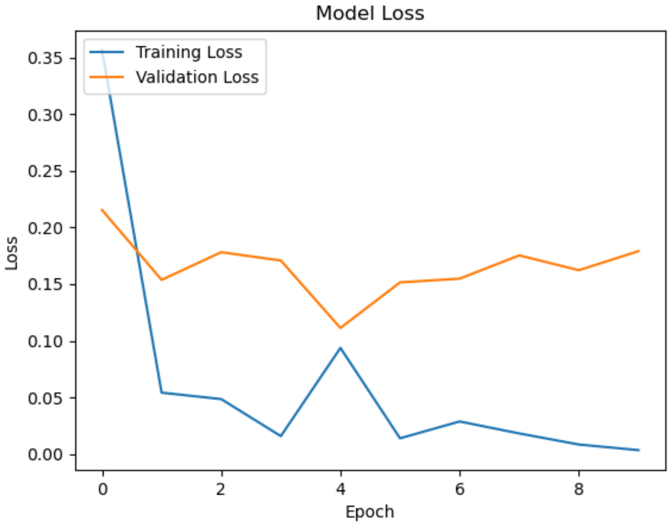


Fig. 7 Training and Validation Loss per Epoch.

DOI: 10.17148/IJARCCE.2026.154285

B. *Discussion*

The use of convolutional layers along with the dropout technique has made feature extraction and generalization effective. The results obtained from this model prove the effectiveness of using CNNs for classifying wildlife images, with data preprocessing being an important aspect. The addition of augmented testing data is another reason behind obtaining accurate results from our model.

V. CONCLUSION

This study successfully developed a CNN for binary classification of bear and panda images, integrating core digital image processing techniques to improve model accuracy and robustness. The approach demonstrates practical utility in ecological monitoring and automated wildlife image analysis. Future directions include expanding the dataset, applying advanced augmentation, and exploring transfer learning to further improve performance.

VI. FUTURE WORK

- Incorporate data augmentation methods such as color jitter and Gaussian noise to simulate environmental variations.
- Experiment with deeper CNN architectures or pretrained models (e.g., ResNet, EfficientNet) for feature transfer.
- Deploy the model in field applications for real-time wildlife monitoring and conservation efforts.
- Explore multi-class classification including additional species for broader ecological studies.

REFERENCES

[1]. Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. Journal of Big Data, 6(1), 1–48. https://doi.org/10.1186/s40537-019-0197-0

[2]. Tan, M., & Le, Q. V. (2019). EfficientNet: Rethinking model scaling for convolutional neural networks. Proceedings of ICML 2019. https://arxiv.org/abs/1905.11946

[3]. Stančić, A., Vyroubal, V., & Slijepčević, V. (2022). Classification efficiency of pre-trained deep CNN models on camera trap images. Journal of Imaging, 8(2), 20. https://doi.org/10.3390/jimaging8020020

[4]. Norouzzadeh, M. S. et al. (2018). Insights and approaches using deep learning to classify wildlife. PLOS ONE / PMC6544615. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6544615/

[5]. Yang, S. et al. (2022). Image data augmentation for deep learning: A survey. arXiv:2204.08610. https://arxiv.org/abs/2204.08610

[6]. Mikołajczyk, A., & Grochowski, M. (2018). Data augmentation for improving deep learning in image classification problem. ResearchGate / IEEE. https://www.researchgate.net/publication/325920702

[7]. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. Journal of Machine Learning Research, 15(1), 1929–1958.

[8]. Alzubaidi, L. et al. (2023). Data augmentation in classification and segmentation: A survey and new strategies. PMC9966095. https://pmc.ncbi.nlm.nih.gov/articles/PMC9966095/

[9]. Swanson, A. et al. / Norouzzadeh et al. cited in: Metadata-augmented deep neural networks for wild animal classification. arXiv:2409.04825. https://arxiv.org/pdf/2409.04825

[10]. ACM (2025). Animal species classification using CNN. Proceedings of SEIM 2025. https://dl.acm.org/doi/10.1145/3725899.3725936