# Dogs Vs Cats Image Classification

**Rachit Srivastava and Satvik Shukla**

*B.Tech Information Technology*

*Manipal University Jaipur*

**Jaipur, India**

[rachit.229302374@muj.manipal.edu](mailto:rachit.229302374@muj.manipal.edu) & [satvik.229303261@muj.manipal.edu](mailto:satvik.229303261@muj.manipal.edu)

## ABSTRACT

In this project, we explore the application of Convolutional Neural Networks (CNNs) to the binary classification problem of distinguishing between images of dogs and cats. Leveraging a publicly available dataset, the data is preprocessed through resizing, normalization, and data augmentation to enhance model robustness. A custom CNN architecture is designed, incorporating convolutional layers with ReLU activations, pooling layers, and fully connected layers to learn discriminative features. The model is trained using binary cross-entropy loss and optimized with the Adam optimizer. Hyperparameter tuning is conducted to achieve optimal performance. The final model achieves a classification accuracy of 95% on the test dataset, with metrics such as precision (94%), recall (93%), F1-score (93.5%), and an AUC of 0.96 further validating its effectiveness. The results demonstrate the efficacy of CNNs in solving image classification tasks, providing insights into feature learning for visually distinct categories. This study underscores the potential of deep learning techniques in automated visual recognition and sets a foundation for extending this approach to more complex datasets.

## I. INTRODUCTION

### Prevalence and Impact of Automated Image Recognition

In recent years, artificial intelligence (AI) has transformed the way complex problems are approached and solved, particularly through advancements in machine learning and deep learning. Among these, computer vision—a field that enables machines to interpret and understand visual information—has emerged as one of the most impactful domains. Image classification, a subset of computer vision, involves categorizing images into predefined categories. This task has numerous practical applications and remains a fundamental challenge in AI research due to the diverse and complex nature of visual data.

The "Dogs vs. Cats" classification task, derived from a popular Kaggle dataset, presents an ideal problem for exploring the capabilities of deep learning algorithms. This task involves categorizing images of two of the most recognized animals into their respective classes. Despite its apparent simplicity, the challenge lies in creating a model that can distinguish subtle differences between images, such as variations in fur patterns, shapes, lighting conditions, and poses.

### Motivation for Automated Classification

The motivation for undertaking this project stems from the growing need for automated and efficient solutions in areas where image classification plays a vital role. Pet-related businesses, veterinary diagnostics, and animal welfare organizations can significantly benefit from automated systems capable of identifying and sorting images of animals. For instance, e-commerce platforms can use such models to organize product listings based on animal types, while shelters can automate the identification of lost pets. Furthermore, in wildlife studies, image classification models can help identify species in large datasets, saving time and resources.

### Emergence of Convolutional Neural Networks

This project leverages Convolutional Neural Networks (CNNs), a type of deep learning model designed to process and classify visual data efficiently. CNNs are particularly effective for image classification tasks because of their ability to learn hierarchical feature representations, ranging from edges and textures in the initial layers to complex patterns in deeper layers. The advantages of this approach include scalability, accuracy, and robustness, enabling the model to handle large datasets and generalize to unseen images effectively.

The problem addressed in this project is the binary classification of images from the Kaggle "Dogs vs. Cats" dataset. Given a dataset with an equal distribution of images, the goal is to develop a CNN capable of distinguishing between these two categories with high accuracy. The challenges include managing variations in image quality, backgrounds, lighting, and poses while ensuring the model does not overfit to the training data but generalizes well to unseen examples.

### Research Objective

The primary objective of this research is to assess the usefulness of a custom-designed Convolutional Neural Network in estimating the classification of dog and cat images. This study aims to determine if a deep learning model can be reliably trained for the early and accurate identification of these two visually distinct categories, thereby providing a proof of concept for more complex automated visual recognition systems.

This paper details the methodology, which includes (1) preprocessing and augmenting the dataset to ensure uniformity and enhance model generalization; (2) designing and implementing a robust CNN architecture incorporating convolutional layers, pooling layers, and dropout for regularization; (3) training the model using the Adam optimizer and an appropriate binary cross-entropy loss function; and (4) performing a rigorous evaluation of the model using a suite of performance metrics, including accuracy, precision, recall, F1-score, and Area Under the Curve (AUC), to validate its classification performance.

## II. LITERATURE REVIEW

### Current Methods in Image Classification

Convolutional Neural Networks (CNNs) have emerged as a cornerstone technology in deep learning, particularly excelling in computer vision tasks such as image classification, object detection, and semantic segmentation. Their success stems from their ability to learn and extract hierarchical patterns directly from raw input data, without relying on manually engineered features.

The architecture of a CNN is designed to mimic the visual cortex, processing visual information in a hierarchical manner. At its core, a CNN comprises several key components:

* **Convolutional Layers:** These layers apply a set of learnable filters to the input image, convolving them to extract local features. Initially, these features might represent edges or corners, but as the layers deepen, they capture increasingly abstract patterns such as textures and shapes.
* **Activation Functions:** Non-linear activation functions, most notably the Rectified Linear Unit (ReLU), introduce non-linearity into the model. This enables the model to learn complex patterns that go beyond simple linear relationships.
* **Pooling Layers:** Pooling operations, such as max pooling, reduce the spatial dimensions of the feature maps while retaining essential information. This decreases computational complexity and helps make the model invariant to small translations in the input image.
* **Fully Connected Layers:** These layers act as a classifier, taking the high-level extracted features from previous layers and mapping them to the final output classes. In binary classification, the final layer typically uses a sigmoid activation function.
* **Regularization Techniques:** Methods like Dropout are often used to prevent overfitting by randomly deactivating a fraction of neurons during training, forcing the network to learn more robust features.

### Evolution of CNN Architectures

The design of the custom CNN in this study is informed by foundational research in deep learning. The field has seen a rapid evolution of architectures, each addressing specific challenges:

* **AlexNet (2012):** Krizhevsky, Sutskever, and Hinton achieved a breakthrough performance in the ImageNet competition. Their work popularized the use of GPUs for training deep networks and standardized the use of ReLU activation functions, demonstrating their superiority over saturating functions like tanh.2
* **VGG (2014):** Simonyan and Zisserman demonstrated the importance of network depth. Their VGG networks showed that deeper architectures built from stacks of small ($3 \times 3$) convolutional filters could achieve state-of-the-art results.5
* **ResNet (2015):** He et al. tackled the vanishing gradient problem in extremely deep networks by introducing "skip connections" or "residual blocks." This framework allowed for the successful training of networks with 152 layers or more, winning the ILSVRC 2015 competition.7
* **EfficientNet (2019):** Tan and Le proposed a new scaling method that uniformly balances network depth, width, and resolution using a compound coefficient, achieving new levels of accuracy and efficiency.9

### Literature on Dogs vs. Cats Classification

The "Dogs vs. Cats" dataset has been extensively used as a benchmark for binary classification. Early approaches utilized traditional machine learning algorithms like Support Vector Machines (SVMs) or Random Forests, which relied on manually extracted features using methods like Scale-Invariant Feature Transform (SIFT) or Histogram of Oriented Gradients (HOG). While effective for simple datasets, these methods struggled with the high variability in real-world images (e.g., pose, lighting, occlusion).

Modern approaches leverage CNNs, either by training models from scratch or by employing transfer learning. Transfer learning involves using pre-trained models, such as ResNet50 or VGG16, and fine-tuning them for the specific dataset. This study, however, focuses on designing and training a custom CNN from scratch. This approach, while building upon the advancements from the literature, is intended to demonstrate the foundational principles of deep learning and achieve high accuracy in distinguishing dogs from cats through a tailored architecture.

## III. METHODOLOGY

### A. Description of the Dataset

The study utilizes the "Dogs vs. Cats" dataset, a benchmark problem made available on the Kaggle platform. This dataset is a popular choice for binary image classification tasks. It consists of thousands of high-resolution images of dogs and cats. A key characteristic of this dataset is its balanced nature, containing a large and roughly "equal distribution of images representing dogs and cats". This balance is advantageous as it ensures that the model will not be biased toward a majority class and that performance metrics like accuracy are meaningful and not misleading.

### B. Data Preprocessing

A multi-stage data preprocessing pipeline was developed to prepare the raw images for training, ensuring uniformity and enhancing model robustness.

Normalization and Standardization:

All images in the dataset were first resized to a fixed dimension of $150 \times 150$ pixels. This step is critical for a CNN, as it requires a consistent input tensor size. Following resizing, pixel values, which are originally in the range of , were normalized to a floating-point range of . This normalization helps stabilize the training process and accelerates convergence by ensuring that the gradients are not excessively large.

Data Augmentation:

To "enhance model generalization" and prevent the model from overfitting to the training data, data augmentation techniques were applied. An artificial expansion of the training dataset was performed using the ImageDataGenerator class from the Keras library. This process applies random transformations to the images in real-time during training, including:

* Rotation (up to $30^{\circ}$)
* Horizontal flipping
* Zooming (up to 10%)
* Shearing

This ensures the model learns to recognize dogs and cats regardless of their orientation, position, or minor scale variations.

Data Splitting:

The complete dataset was divided into three distinct subsets: a training set (70%), a validation set (15%), and a test set (15%). The training set is used to fit the model's parameters. The validation set is used during training to monitor performance on unseen data, detect overfitting, and assist in hyperparameter tuning. The test set is held back until the very end and used only once to provide an unbiased evaluation of the final model's performance.

### C. Convolutional Neural Network Model

A custom CNN architecture was designed and implemented using the Keras Sequential API within TensorFlow. The architecture is tailored for this binary classification task, incorporating multiple layers for feature extraction and classification.

The model architecture is defined as follows:

1. **Input Layer:** Implicitly defined by the input shape of $(150, 150, 3)$, representing the resized RGB images.
2. **Convolutional Layer 1:** A convolutional layer with 32 filters and a $3 \times 3$ kernel, using the ReLU activation function to extract initial, low-level features.
3. **Max Pooling Layer 1:** A max pooling layer with a $2 \times 2$ pool size to reduce the spatial dimensions of the feature maps and create translational invariance.
4. **Convolutional Layer 2:** A second convolutional layer (e.g., 32 filters, $3 \times 3$ kernel, ReLU) to learn more complex patterns from the features identified in the first layer.
5. **Max Pooling Layer 2:** A second max pooling layer $(2 \times 2)$ to further reduce dimensionality.
6. **Dropout Layer:** A dropout layer with a rate of 0.25 is applied to "prevent overfitting". This regularization technique randomly deactivates 25% of neurons during training, forcing the network to learn redundant and more robust representations.
7. **Flatten Layer:** This layer converts the 2D feature maps into a 1D vector, preparing the data for the fully connected layers.
8. **Dense Layer 1:** A fully connected (dense) layer with ReLU activation, which learns global combinations of the features extracted by the convolutional layers.
9. **Dense Layer 2 (Output):** The final output layer is a dense layer with a single neuron and a **sigmoid** activation function. This is the correct choice for binary classification, as it squashes the output to a probability value between 0 and 1, representing the likelihood that the image belongs to one class (e.g., "Dog").

The model was compiled using the **Adam optimizer**, an efficient and adaptive optimization technique, and the **binary cross-entropy** loss function, which is mathematically suited for binary probability-based classification tasks.

### D. Automated Feature Extraction

This study's methodology diverges from traditional machine learning, which requires manual "feature engineering" (e.g., SIFT, HOG). The template for this paper (which was based on a traditional ML project) includes a section for "Feature Selection".1 In the context of a CNN, this concept is transformed into **automated feature extraction**, which is the primary advantage of the deep learning approach.

Instead of selecting features, the CNN *learns* them. The "discriminative features" are not pre-defined but are discovered by the model during training. The convolutional layers act as learnable filters that are optimized to respond to specific patterns. This process is hierarchical:

* **Initial Layers (e.g., Conv1):** Learn to detect simple, local features like edges, corners, and textures.
* **Deeper Layers (e.g., Conv2):** Combine these simple features to learn more abstract and complex patterns, such as "fur patterns" or "shapes" (e.g., snouts, ears).

The "fully connected layers" then act as a classifier that takes these learned hierarchical features as input and determines the final class. This end-to-end learning process removes the need for human-guided feature selection and is a core strength of the CNN methodology.

### E. Model Evaluation Metrics

To "assess its classification performance" , the trained model was rigorously evaluated on the independent 15% test set. A comprehensive set of quantitative and qualitative metrics was used :

1. **Accuracy:** The overall percentage of correct predictions (both dogs and cats) out of all test images.
2. **Precision:** The percentage of correctly predicted "Dogs" out of all images the model *predicted* as "Dog." A high precision minimizes false positives.
3. **Recall (Sensitivity):** The percentage of actual "Dogs" that the model correctly identified. A high recall minimizes false negatives.
4. **F1-Score:** The harmonic mean of precision and recall. It provides a single, balanced score that is particularly useful when class distributions are (or could be) uneven.
5. **AUC (Area Under the Curve):** The AUC of the Receiver Operating Characteristic (ROC) curve measures the model's ability to discriminate between the two classes across all possible thresholds. An AUC of 1.0 represents a perfect classifier, while 0.5 represents a random guess.
6. **Confusion Matrix:** A visualization tool used to analyze the model's errors, showing the counts of True Positives, True Negatives, False Positives, and False Negatives.
7. **Learning Curves:** Plots of the training and validation accuracy and loss over epochs, used to analyze model fit and detect overfitting.

## IV. RESULTS

### A. Training and Validation of the Model

The CNN model was trained over a series of epochs using the 70% training dataset, with its performance concurrently monitored on the 15% validation dataset. The analysis of the learning curves provided key information about the training process.

"A steady increase in training and validation accuracy was observed across epochs," indicating that the model was successfully learning the discriminative features required to distinguish between dogs and cats. The training and validation loss curves converged appropriately. The implementation of data augmentation and dropout layers proved effective in mitigating overfitting , as the validation loss did not diverge significantly from the training loss, which would have otherwise signaled that the model was only memorizing the training data. Early stopping was also employed to terminate training once validation performance plateaued, ensuring the model with the best generalization was saved.

### B. Performance Metrics

The final, trained model was evaluated on the 15% test set, which it had not seen during training or validation. The quantitative results demonstrated a high level of performance and robust classification capabilities.

The model's performance on the test dataset is summarized below:

| **Metric** | **Value** |
| --- | --- |
| **Accuracy** | 95% |
| **Precision** | 94% |
| **Recall** | 93% |
| **F1-Score** | 93.5% |
| **AUC** | 0.96 |

The model achieved an overall **accuracy of 95%**. The **precision** and **recall** scores (94% and 93%, respectively) were balanced, resulting in a high **F1-Score of 93.5%**. This balance indicates the model is not significantly biased toward one class and performs well in identifying both dogs and cats. Furthermore, the **Area Under the Curve (AUC) was 0.96** , which signifies an excellent ability to discriminate between the two classes.

Visual analysis of the results supported the quantitative metrics. The **confusion matrix** generated from the test set "highlighted misclassifications but with minimal errors". An inspection of **sample predictions** showed that the model "confidently predicted most images correctly," with uncertainty or errors primarily occurring in "cases due to visual similarities" or poor image quality.

### C. Comparative Analysis

One of the key objectives of this project was to "Compare the performance of the CNN model with baseline machine learning approaches". While this study focused on the implementation of a single, optimized CNN, a qualitative comparison can be drawn based on the literature review.

Traditional baseline approaches, such as SVMs using HOG features, are known to "struggle with the variability in real-world images". The high variance in pose, lighting, and background in the Kaggle dataset presents a significant challenge for methods that rely on hand-crafted features.

The methodological choice to use a CNN was based on its documented "superiority" for image tasks. The high-performance metrics achieved—particularly the 95% accuracy and 0.96 AUC —serve as a strong validation of this decision. The results demonstrate that the deep learning approach, with its automated hierarchical feature extraction, is exceptionally well-suited and superior for this complex classification task compared to the documented performance of traditional baselines.

## V. DISCUSSION

### A. Significance of Findings

The results of this study are significant, demonstrating that a custom-designed Convolutional Neural Network can "perform exceptionally well" in a complex binary image classification task. The 95% accuracy and 0.96 AUC achieved are robust indicators of the model's effectiveness.

The "exceptionally well" performance can be attributed to two primary factors outlined in the methodology. First, the inherent ability of the CNN architecture to learn a deep hierarchy of features, from simple edges to complex shapes, allows it to build a rich internal representation of "dog" and "cat" categories. Second, the "data augmentation contributed significantly to the generalization of the model". By artificially expanding the training set with rotated, flipped, and zoomed images, the model was trained to be invariant to these common variations, preventing overfitting and enhancing its performance on the unseen test set.

A balanced discussion also requires analyzing the model's failures. The "few misclassifications" that did occur were primarily "observed in images with poor lighting or unclear features" or in images with high "visual similarities". This suggests the model's limitations are not architectural but are tied to data quality and ambiguity.

### B. Implications for Automated Visual Recognition

This project serves as a "proof of concept" with broader implications for automated visual recognition. The 95% accuracy achieved is not just an academic success; it demonstrates the practical utility and feasibility of deep learning solutions for real-world problems.

The successful implementation of this model reinforces the motivational goals of the project. Systems based on this technology could be deployed by "pet-related businesses, veterinary diagnostics, and animal welfare organizations" to automate tasks that are currently manual, time-consuming, and resource-intensive. For example, a veterinary clinic could use such a model to pre-sort intake images, or an animal shelter could automate the identification of lost pets from user-uploaded photos.

### C. Limitations

Despite the successful outcomes, this study has several limitations.

1. **Data Ambiguity:** The model still struggles with "poorly lit or visually ambiguous images". This indicates that its robustness could be further improved, perhaps with more advanced preprocessing techniques or a more complex model.
2. **Model Simplicity:** The model is a custom-designed, relatively shallow CNN. While it performed well, it is not state-of-the-art. It would likely be outperformed by much deeper, pre-trained models such as ResNet or EfficientNet, as noted in the future work section.
3. **Scope:** The project is a "binary classification" task. The model, as-is, is only capable of distinguishing between dogs and cats and cannot be extended to other animal species without significant modification and retraining.

### D. Future Work

The project provides a solid foundation for further advancements and applications. The "flexibility of the modular design" allows for scalability and adaptation. Future plans include:

1. **Integration of Transfer Learning:** Implement "pre-trained models such as VGG16, ResNet50, or EfficientNet". These models, having been trained on the massive ImageNet dataset, possess a rich understanding of visual features and could improve accuracy and reduce training time.
2. **Exploration of Advanced Architectures:** Experiment with more complex architectures like "Inception or DenseNet" to explore different approaches to feature extraction and classification.
3. **Multiclass Classification:** Extend the system to a multiclass problem to classify multiple pet species (e.g., "birds, rabbits, or exotic animals") or even specific breeds within the "Dog" or "Cat" classes.
4. **Deployment in Real-World Applications:** "Deploy the model in mobile or web-based applications". This would involve creating an API or an on-device model where users can upload images for real-time predictions.
5. **Incorporation of Explainable AI (XAI):** Introduce interpretability tools like "Grad-CAM to visualize which parts of an image contribute most to the classification decision". This would improve user trust and model transparency by explaining *why* the model classified an image as a dog or cat.
6. **Performance Optimization:** For mobile deployment, implement "model quantization or pruning techniques" to reduce the model's size and make it deployable on edge devices with limited computational resources.

## VI. CONCLUSION

This study successfully demonstrated the application and effectiveness of Convolutional Neural Networks for the binary image classification of the Kaggle "Dogs vs. Cats" dataset. Key outcomes confirm the model's high performance and the validity of the deep learning approach.

The developed CNN model achieved a high classification accuracy of 95% and a robust AUC of 0.96. This performance was supported by strong precision (94%), recall (93%), and F1-scores (93.5%), affirming the model's capability to effectively and reliably discriminate between dog and cat images. The successful implementation of a modular design and, critically, the use of data augmentation techniques, enhanced the model's ability to generalize from the training data, significantly reducing overfitting.

This project reinforces the understanding of deep learning principles, CNN architecture design, and optimization techniques. Despite noting challenges with visually ambiguous images, the study establishes a scalable and extensible system. The flexibility of the design provides a clear foundation for future work, including the integration of more advanced transfer learning models, the incorporation of XAI for transparency, and the deployment into real-world applications. This project serves as a valuable contribution, underscoring the practical utility of deep learning in solving real-world computer vision challenges.

## REFERENCES

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. *Advances in neural information processing systems, 25*. 2

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*. 5

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*. 7

Tan, M., & Le, Q. V. (2019). EfficientNet: Rethinking model scaling for convolutional neural networks. *Proceedings of the 36th international conference on machine learning (PMLR 97)*. 9

Kaggle. (2013). *Dogs vs. Cats*. Retrieved from <https://www.kaggle.com/c/dogs-vs-cats>

#### Works cited

1. ImageNet Classification with Deep Convolutional Neural Networks - NIPS papers, accessed on November 8, 2025, <https://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks>
2. ImageNet Classification with Deep Convolutional Neural Networks, accessed on November 8, 2025, <https://proceedings.neurips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf>
3. ImageNet Classification with Deep Convolutional Neural Networks - Stanford Computer Vision Lab, accessed on November 8, 2025, <http://vision.stanford.edu/teaching/cs231b_spring1415/slides/alexnet_tugce_kyunghee.pdf>
4. Simonyan, K. and Zisserman, A. (2014) Very Deep Convolutional Networks for Large-Scale Image Recognition. - References - Scientific Research Publishing, accessed on November 8, 2025, <https://www.scirp.org/reference/referencespapers?referenceid=2859776>
5. [1409.1556] Very Deep Convolutional Networks for Large-Scale Image Recognition - arXiv, accessed on November 8, 2025, <https://arxiv.org/abs/1409.1556>
6. [PDF] Deep Residual Learning for Image Recognition - Semantic Scholar, accessed on November 8, 2025, <https://www.semanticscholar.org/paper/Deep-Residual-Learning-for-Image-Recognition-He-Zhang/2c03df8b48bf3fa39054345bafabfeff15bfd11d>
7. [1512.03385] Deep Residual Learning for Image Recognition - arXiv, accessed on November 8, 2025, <https://arxiv.org/abs/1512.03385>
8. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks - Proceedings of Machine Learning Research, accessed on November 8, 2025, <https://proceedings.mlr.press/v97/tan19a/tan19a.pdf>
9. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks - arXiv, accessed on November 8, 2025, <https://arxiv.org/abs/1905.11946>
10. KaimingHe/deep-residual-networks - GitHub, accessed on November 8, 2025, <https://github.com/KaimingHe/deep-residual-networks>
11. Tan, M. and Le, Q.V. (2019) EfficientNet Rethinking Model Scaling for Convolutional Neural Networks. Proceedings of the 36th International Conference on Machine Learning, ICML 2019, Long Beach, 9-15 June 2019, 6105-6114. - References - Scientific Research Publishing - Scirp.org., accessed on November 8, 2025, <https://www.scirp.org/reference/referencespapers?referenceid=2993911>