

# CNN Cat vs Dog Classifier



Code: <https://github.com/annimukherjee/cats-vs-dogs-cnn>

## Problem Statement

The objective of this project is to design and train a **Convolutional Neural Network (CNN)** that can classify images of cats and dogs. This is a binary classification problem where the model must output whether the given input image belongs to the "Cat" class or the "Dog" class.

## Dataset

- Source: Kaggle's Cats vs Dogs dataset ( `./dataset/` ).
- Structure:
  - **training\_set/** → used for training & validation with an **80/20 split**
  - **test\_set/** → unseen images for model evaluation
- Preprocessing:
  - Images resized to **128×128 pixels**.
  - **Data Augmentation**: Random flips, rotations, and zoom applied on training data.
  - **Normalization**: Pixel values scaled to range `[0, 1]` .

## Approach

### 1. Model Architecture:

- Input Layer (128×128×3)
- Conv2D → ReLU → MaxPooling
- Conv2D → ReLU → MaxPooling
- Flatten → Dense(128, ReLU) → Dense(1, Sigmoid)
- Total: ~ 3,696,801 parameters.

## 2. Training:

- Optimizer: Adam
- Loss: Binary Crossentropy
- Metrics: Accuracy
- Epochs: 25
- Batch Size: 32

## 3. Evaluation:

- Performance measured on validation and independent test set.
- Visualization of **training/validation accuracy and loss curves**.
- Random test image predictions with predicted vs. actual labels.

## Results

- **Test Accuracy:** 77.56%
- Predictions on unseen images were largely correct, with occasional misclassifications in edge cases (ambiguous images).

## Challenges & Learnings

- **Data Quality:** Some images contained noise or ambiguous labeling, which affected predictions.
- **Overfitting:** Initially, the model showed signs of overfitting; adding **data augmentation** and normalization improved generalization.
- **Key Learning:** Well-structured preprocessing and augmentation can significantly enhance model performance on small to medium datasets.

---

## References & Inspiration

- [Kaggle Cats vs Dogs Tutorial](#)
- [Becoming Human Blog](#)
- [CampusX YouTube CNN Tutorial](#)