Report on Cat and Dog Image Classification Using CNN

1. Introduction

The goal of this project was to build a Convolutional Neural Network (CNN) to classify images of cats and dogs using the dataset available on Kaggle. This report outlines the methodology, implementation, and performance evaluation of the image classification model.

2. Dataset Description

The dataset, sourced from Kaggle (Dataset Link), contains labeled images of cats and dogs. It is divided into training, validation, and test sets to evaluate the model's performance accurately. The dataset was preprocessed to normalize pixel values and resize all images to 150x150 pixels for consistency.

- Training set: Contains images with data augmentation applied to improve model robustness.
- Validation set: Used to monitor the model's performance during training.
- Test set: Evaluates the final performance of the trained model.

3. Methodology

3.1 Data Preprocessing

- The dataset was unzipped and organized into appropriate directories for training and testing.
- Data augmentation techniques (rotation, width/height shift, horizontal flipping) were applied to the training set to enhance diversity.
- Pixel values were normalized to the range [0, 1].

3.2 Model Architecture The CNN was designed with the following architecture:

- Input: 150x150 RGB images
- Convolutional Layers:
    - Three blocks of Conv2D + BatchNormalization + MaxPooling2D
    - Filters: 32, 64, and 128
    - Activation: ReLU
    - Regularization: L2 with weight decay
- Flatten Layer: Converts feature maps to a single vector.
- Fully Connected Layers:
    - Dense(256) with ReLU and Dropout (40%)

- Dense(1) with Sigmoid activation for binary classification

- Total Parameters: ~10.7M

Model Summary:

Model: "sequential_7"

Layer (type)	Output Shape	Param #
conv2d_21 (Conv2D)	(None, 150, 150, 32)	896
batch_normalization_8 (BatchNormalization)	(None, 150, 150, 32)	128
max_pooling2d_21 (MaxPooling2D)	(None, 75, 75, 32)	0
conv2d_22 (Conv2D)	(None, 75, 75, 64)	18,496
batch_normalization_9 (BatchNormalization)	(None, 75, 75, 64)	256
max_pooling2d_22 (MaxPooling2D)	(None, 37, 37, 64)	0
conv2d_23 (Conv2D)	(None, 37, 37, 128)	73,856
batch_normalization_10 (BatchNormalization)	(None, 37, 37, 128)	512
max_pooling2d_23 (MaxPooling2D)	(None, 18, 18, 128)	0
flatten_7 (Flatten)	(None, 41472)	0
dense_14 (Dense)	(None, 256)	10,617,088
batch_normalization_11 (BatchNormalization)	(None, 256)	1,024
dropout_7 (Dropout)	(None, 256)	0
dense_15 (Dense)	(None, 1)	257

Total params: 10,712,513 (40.86 MB)
Trainable params: 10,711,553 (40.86 MB)
Non-trainable params: 960 (3.75 KB)

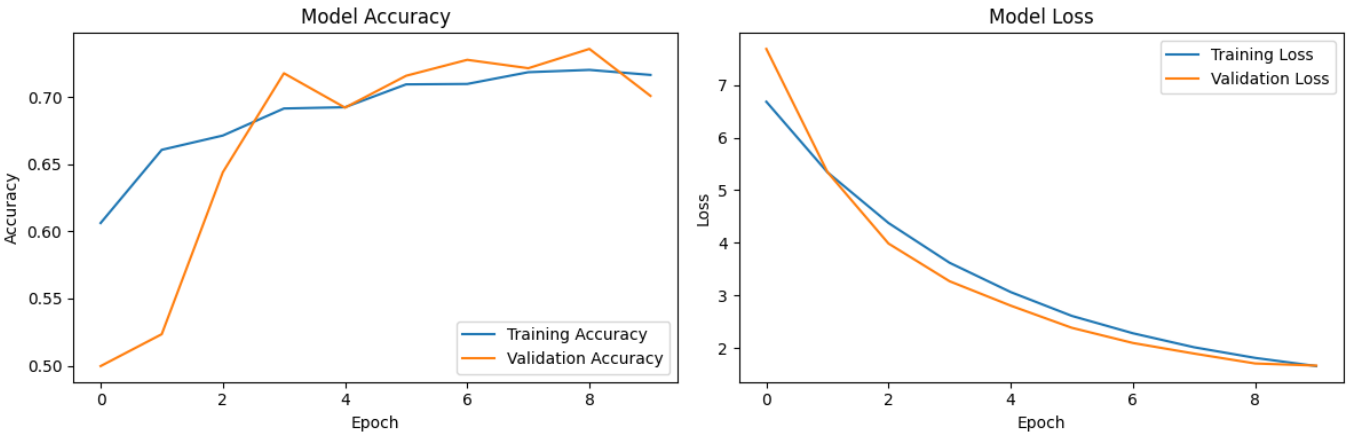
3.3 Training Process

- Optimizer: Adam with a learning rate of 1e-4.
- Loss Function: Binary Crossentropy.
- Metrics: Accuracy.
- Callbacks:
    - EarlyStopping: Monitored validation loss and stopped training after 5 epochs of no improvement.
    - ReduceLROnPlateau: Reduced learning rate by a factor of 0.2 if validation loss plateaued for 3 epochs.
- Number of Epochs: 10.

4. Results

4.1 Training Performance The model was trained for 10 epochs with consistent improvement in validation accuracy. Training and validation loss decreased steadily, indicating effective learning without significant overfitting.

- Final Training Accuracy: ~71%
- Final Validation Accuracy: ~74%



4.2 Test Performance The model was evaluated on the test set to determine its generalization capability.

- Test Accuracy: 71%
- Test Loss: 0.49

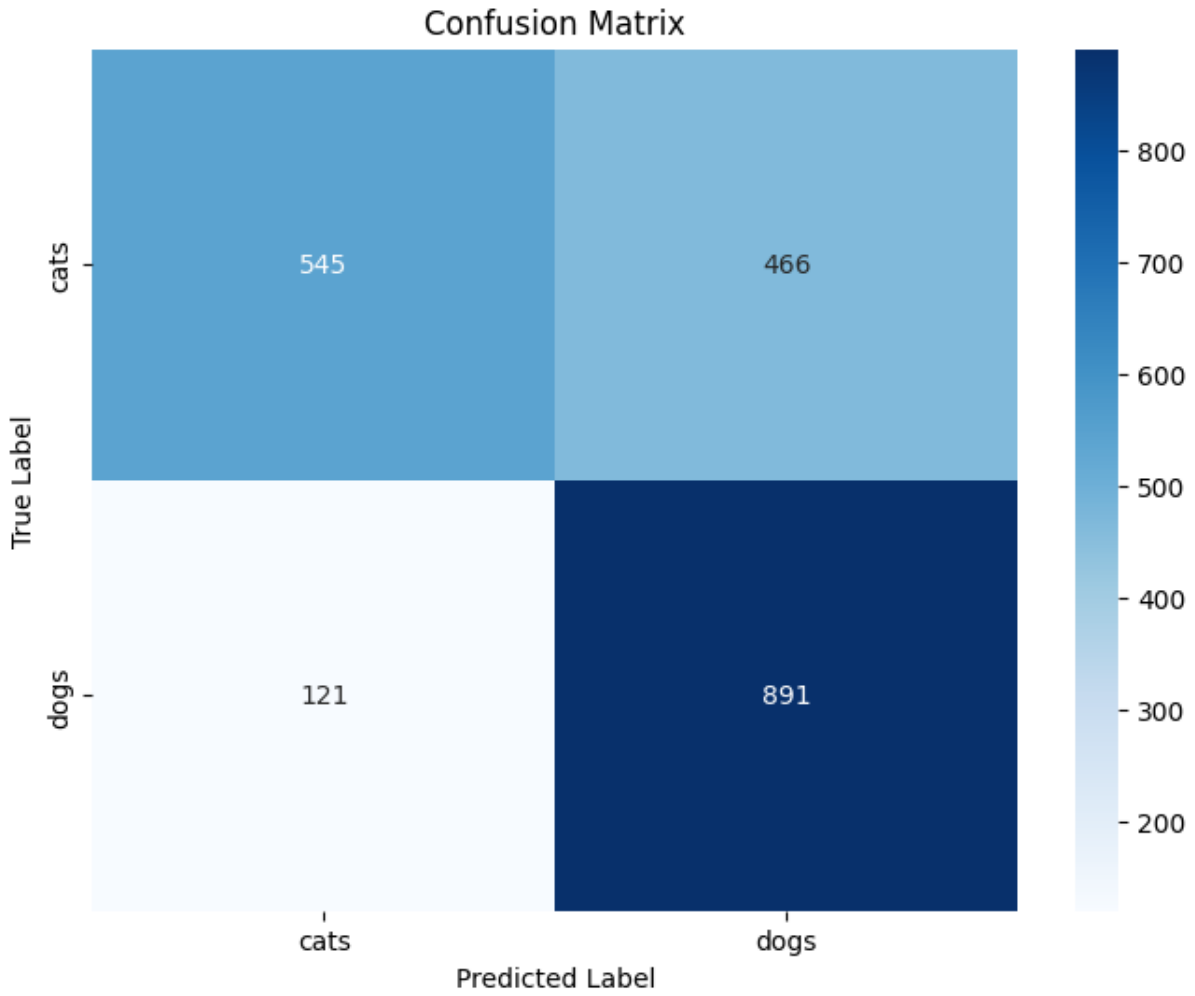
Test Dataset Information:

- Number of test samples: 2023
- Class indices: {'cats': 0, 'dogs': 1}

Prediction Summary:

- Number of predictions: 2023
- Distribution of predictions: [666 cats, 1357 dogs]
- Distribution of true labels: [1011 cats, 1012 dogs]

4.3 Confusion Matrix The confusion matrix showed the distribution of true positives, true negatives, false positives, and false negatives, providing insights into model performance across classes.

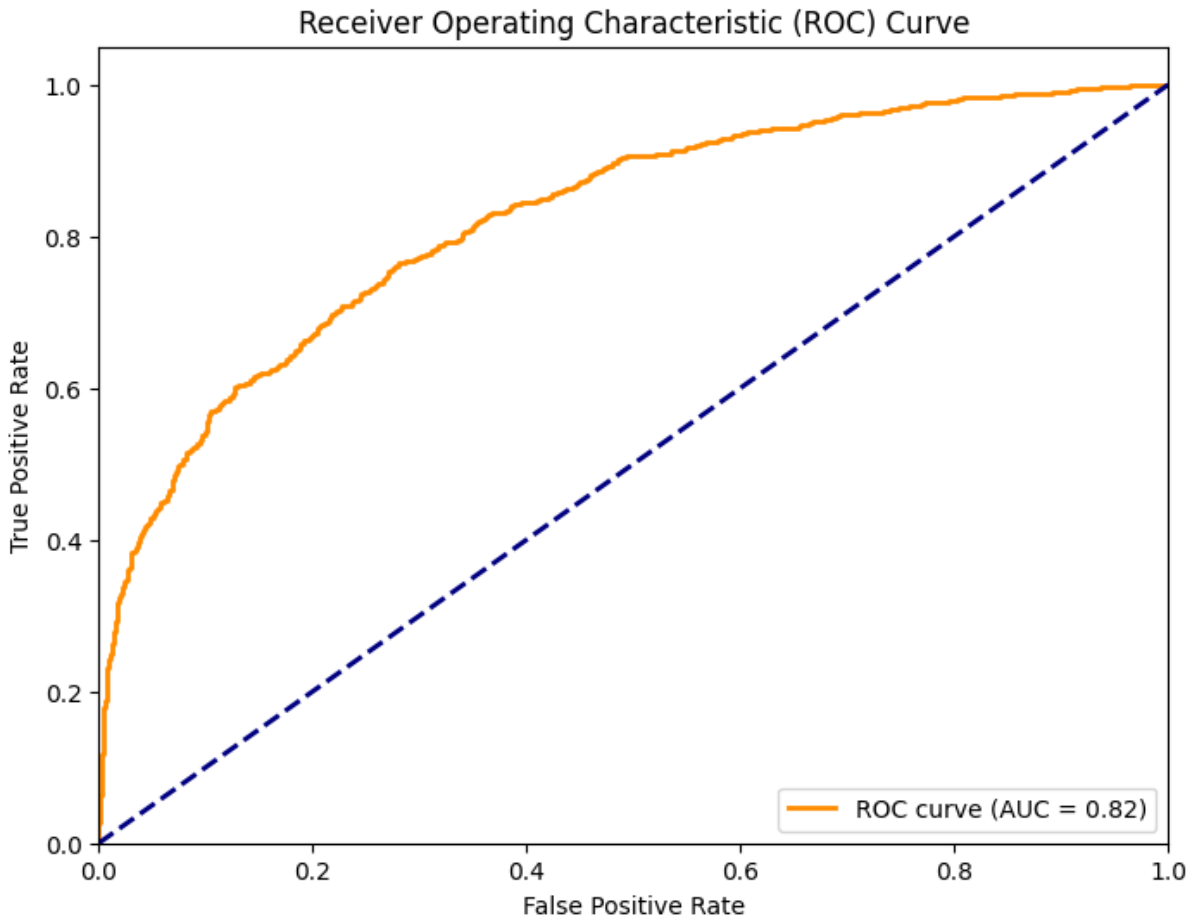


4.4 Classification Report

	precision	recall	f1-score	support
cats	0.82	0.54	0.65	1011
dogs	0.66	0.88	0.75	1012
accuracy			0.71	2023
macro avg	0.74	0.71	0.70	2023
weighted avg	0.74	0.71	0.70	2023

4.5 ROC Curve and AUC

The ROC curve demonstrated the balance between sensitivity and specificity.



5. Discussion

5.1 Key Observations

- Data augmentation significantly improved model robustness and reduced overfitting.
- The model demonstrated consistent accuracy on both validation and test sets, confirming its ability to generalize.

5.2 Limitations

- The model's performance may vary with more complex datasets.
- Further hyperparameter tuning and deeper architectures could potentially improve accuracy.

5.3 Future Work

- Experimenting with transfer learning using pre-trained models like VGG16 or ResNet.
- Increasing dataset size or applying synthetic data generation techniques.

6. Conclusion

This project successfully implemented a CNN to classify cat and dog images with a test accuracy of 71%. While the model achieved satisfactory performance, future work could explore more advanced techniques to enhance accuracy further. The results demonstrate the feasibility of using CNNs for binary image classification tasks.

7. Links

- Kaggle Dataset: https://www.kaggle.com/datasets/tongpython/cat-and-dog/data
- Colab Notebook: https://colab.research.google.com/drive/1t0_tO4ytB6KJy1VqoMA18jz5R-GIrvL-
- Model, python and result files: https://drive.google.com/drive/folders/1FkLFrKdJ_SY2NY7vd6LVNmZ2rIspBeMq?usp=sharing