Cat and Dog Image Classification Using CNN

By – Kumuda Valli V & Tien Duong

Introduction

Convolution Neural Network:

- CNN is a well-established model for image classification. It is composed of multiple layers that reduce the original image into feature maps (pooling).
- These feature maps are then used to classify the image.

DenseNet:

- An extension of CNN that enhances propagation through the neural network.
- Dense connectivity: each layer receives inputs from all preceding layers and solves the vanishing gradient problem.
- Can achieve the same or better accuracy with fewer parameters

Dataset

- 32,000 images obtained from Kaggle (both dog and cat images)
- To avoid storage issues, we use the same size image (32x32 pixels)
- Preprocess image by normalizing the pixel value between 0 and 1
- Training dataset is selected randomly to avoid biased training
- Testing dataset is separate from the training dataset. We use the 80/20 split

Models used

Why CNN?

CNNs were chosen for their straightforward implementation and proven effectiveness in image classification tasks. Their ability to capture spatial hierarchies through convolutional layers makes them ideal for processing image data.

Why DenseNet?

DenseNet121 was introduced to improve performance. DenseNet's dense connectivity allows for efficient feature reuse and gradient flow, addressing limitations in deeper CNN architectures.

Pros of CNN

- CNNs are simple and effective in extracting features from images, allowing for straightforward interpretation and implementation in binary classification tasks.
- Hierarchical feature learning
- Translation invariance

Pros of DenseNet

- DenseNet's architecture allows each layer to access the feature maps from all preceding layers, promoting feature reuse. This leads to more efficient learning and reduces the need for redundant computations.
- Improved gradient flow
- Parameter efficiency

Training process

01
Data preprocessing

Both models utilize transformations such as resizing, normalization, and data augmentation techniques like random horizontal flips and rotations to enhance generalization.

02
EDA

Before training, EDA is conducted to understand the dataset's characteristics, including class distribution, pixel intensity histograms, and sample visualizations. This analysis helps identify potential biases or imbalances in the data and informs preprocessing strategies.

03
Training loop

The training loop for both models involves optimizing weights using backpropagation, the RMSprop and Adam optimizers with weight decay, and BCELoss. Early stopping is implemented based on validation loss to ensure the model does not overfit.

04
Epochs

Training involves several epochs with early stopping to prevent overfitting, ensuring efficient training for both architectures. Early stopping monitors validation performance and halts training when improvements plateau.

Evaluation metrics

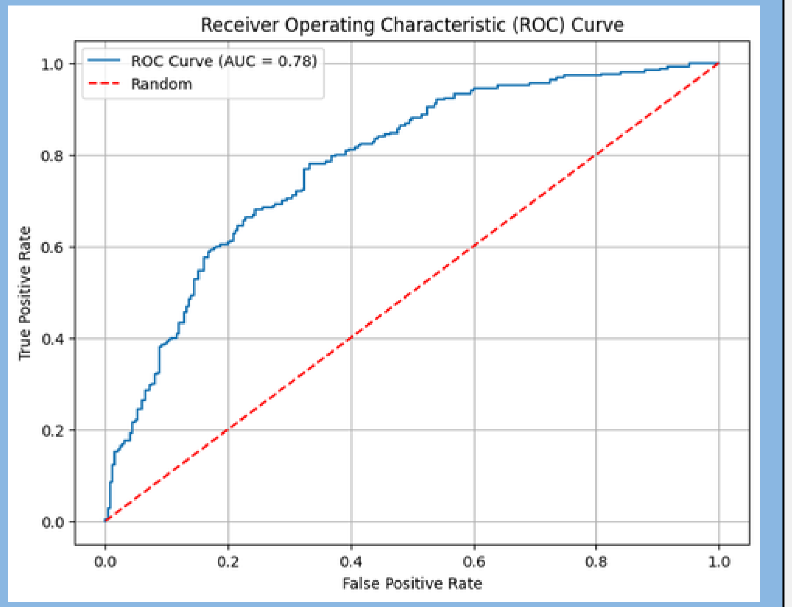
CNN

Accuracy: 71.20%
Precision: 0.68
Recall: 0.80
F1 Score: 0.74
AUC: 0.78

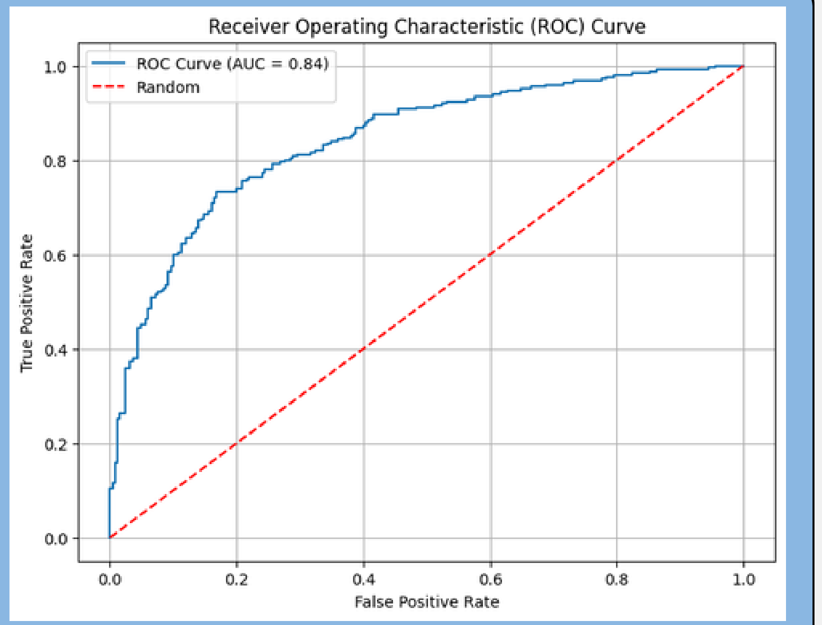
DenseNet

Accuracy: 76.60%
Precision: 0.79
Recall: 0.73
F1 Score: 0.76
AUC: 0.84

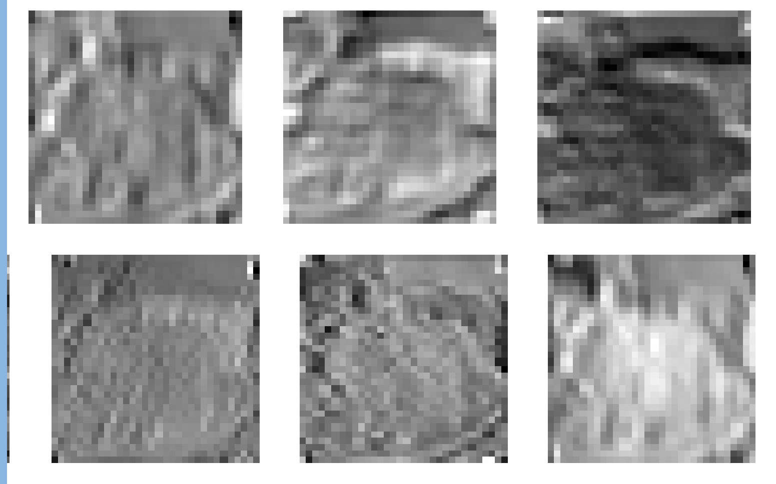
CNN Area Under Curve



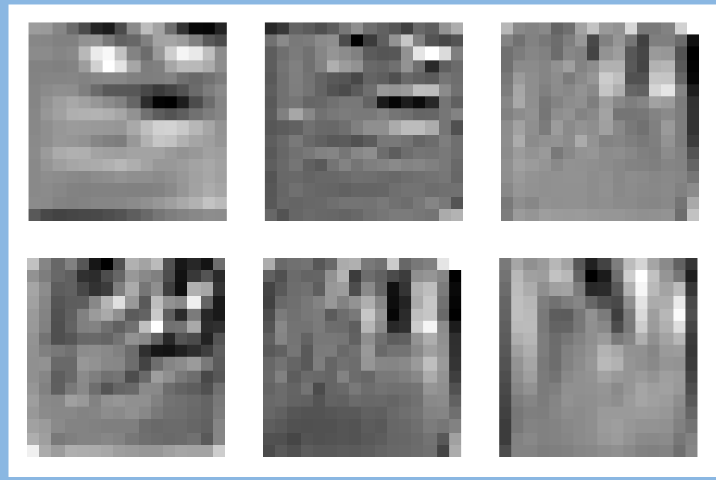
DenseNet Area Under Curve



CNN Feature maps



DenseNet Feature maps



Visualization

Weaknesses

- Small-sized image => fewer features
- Time constraint, hence we didn’t utilize the full dataset
- The epochs was low, which can affect the accuracy

Threats

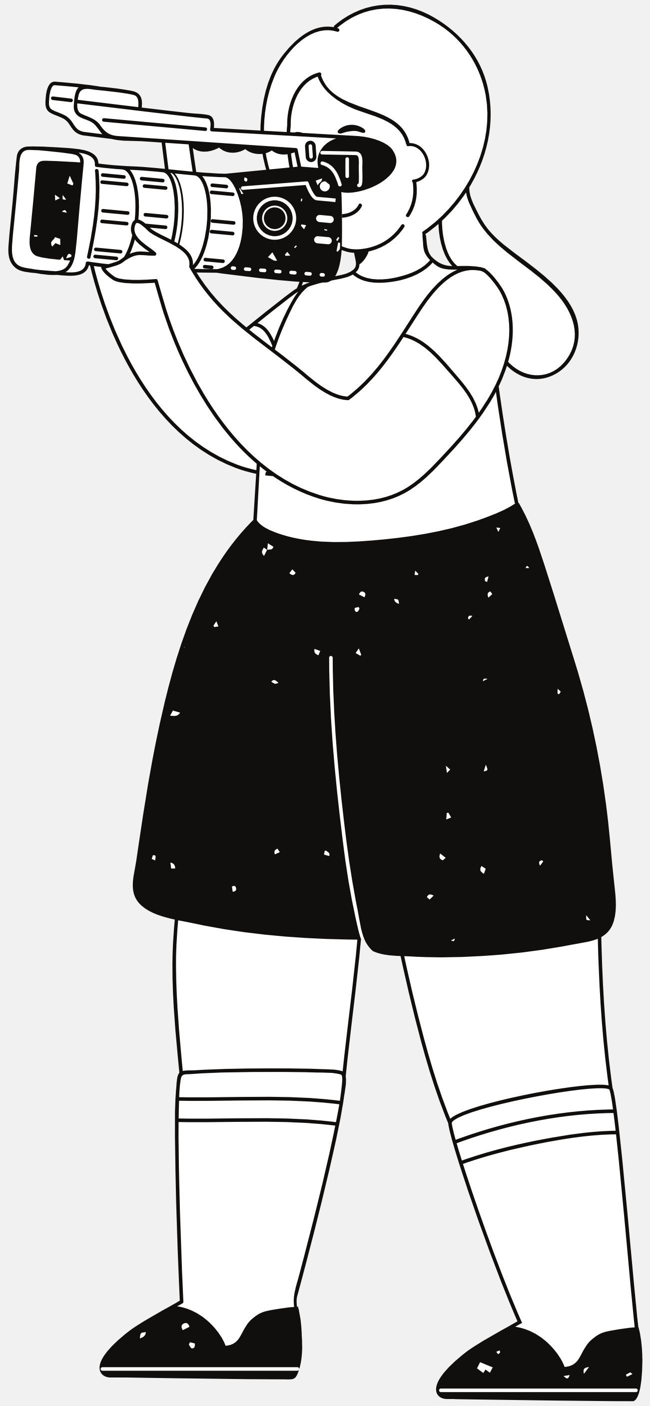
- The model needs to be adjusted properly if used on a larger image
- Lack of handling of overfitting
- Depending on large amount of dataset

Strengths

- Model is user-friendly => good for beginner to learn and understand
- Not requiring high details images
- Low epochs can still lead to good accuracy range

Opportunities

- Transfer learning can be used to classify different animals.
- Other data augmentation such as rotation can be apply to
- Can be expanded to more features image to capture more details.



Conclusion

- Demonstrated the use of CNN in image classification.
- Optimal accuracy and metrics of an amateur model
- Future work can be expanded by transfer learning or a different dataset
- Other CNN architectures can be explored (ResNet, AlexNet)

Thank you