ASSIGNMENT 3

Convolution Networks

This project is about building and comparing several computer models that can tell whether an image shows a cat or a dog. The project uses deep learning, which allows computers to learn patterns in pictures on their own.

In the first part of the project, three models were trained from scratch using different amounts of training images. These models were called Model 1, Model 2, and Model 3. The goal was to see how the size of the training data affects the model's performance and accuracy.

In the second part of the project, another set of three models was created using a technique called transfer learning. These models used a pretrained network called MobileNetV2, which already knew many features of images from a large dataset. The idea was to check if using a pretrained model could improve the results compared to the models that were trained from scratch.

The project includes steps like preparing the data, creating the models, training them, testing their accuracy, and comparing all results. The main aim is to find which model gives the best accuracy and performs well on new unseen images.

DATA OVERVIEW

The dataset used in this project contains pictures of cats and dogs. These images were taken from the popular Kaggle Cats vs Dogs dataset. Each image belongs to one of two classes: Cat or Dog. The dataset was first stored in Google Drive and then organized into separate folders for training, validation, and testing.

For the first part of the project, where the models were trained from scratch, three different training sample sizes were created:

- Model 1 used 1,000 training images (500 cats and 500 dogs)
- Model 2 used 1,300 training images (650 cats and 650 dogs)
- Model 3 used 3,000 training images (1500 cats and 1500 dogs)

Each model also used 500 images for validation and 500 images for testing. The validation set helped monitor how the model performed during training, while the test set was used at the end to check the final accuracy on new unseen images.

In the second part of the project, the same dataset structure was used for transfer learning models. The data was slightly increased or kept the same depending on the experiment, but it always contained a balanced number of cat and dog images.

Before training, the images were resized to a uniform size of 180 × 180 pixels, and pixel values were scaled between 0 and 1 to help the model train efficiently. Data augmentation methods like random flipping, rotation, and zooming were also used to make the model more robust and reduce overfitting.

MODEL TRAINING AND EVALUATION

The models in this project were built and trained using TensorFlow and Keras in Google Colab. The process was divided into two main phases models trained from scratch and models trained using transfer learning.

For Models 1, 2, and 3, the training started from scratch. A Convolutional Neural Network was created with layers such as Conv2D, Batch Normalization, MaxPooling, and Dense layers. Each layer helped the model learn important patterns like edges, shapes, and textures in the images. The pixel values of all images were normalized, and data augmentation was used to improve generalization.

The models were trained for up to 30 epochs, but the actual number depended on early stopping when validation loss stopped improving.

In the next phase, three more models were created using Transfer Learning with the MobileNetV2 pretrained network. The lower layers of MobileNetV2 were frozen so that the model could keep its previously learned features from ImageNet, and only the top layers were trained on the cats and dogs dataset. Later, the top few layers were unfrozen and fine-tuned with a lower learning rate to improve performance further.

The models were compiled using:

Optimizer : Adam

Loss Function : Binary Cross-entropy

Evaluation Metric : Accuracy

Performance Evaluation Metrics

The performance of each model was measured using two main metrics, Accuracy and Loss. Accuracy shows how many images the model predicted correctly out of all the test images.

Loss measures how far the model's predictions are from the correct labels; lower loss means better learning. During training, both training accuracy and validation accuracy were monitored to ensure the model was learning properly without overfitting. The same applied to training loss and validation loss. After training, the models were tested on the test dataset to check their final accuracy. Plots of accuracy and loss across epochs were also used to visually evaluate performance.

In the transfer learning phase, the models achieved higher accuracy compared to the ones trained from scratch, showing that pretrained models can recognize features more effectively even with fewer training images.

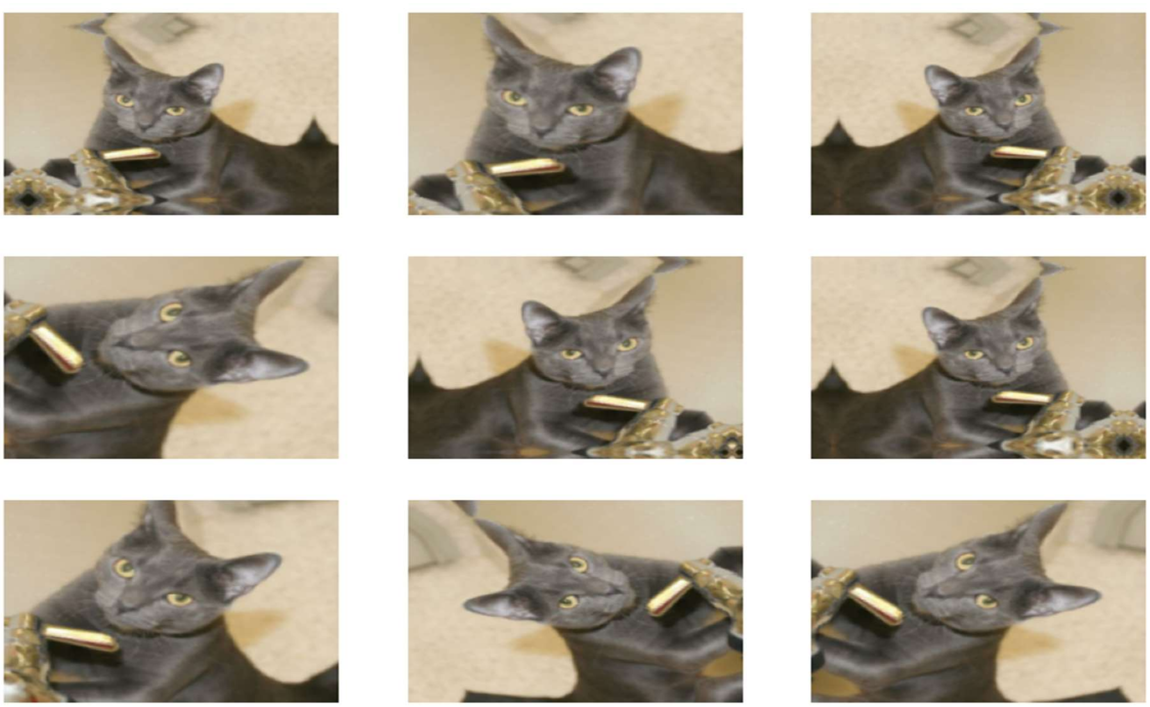
DATA AUGMENTATION

Data augmentation was used to make the models more accurate and prevent overfitting. It creates new versions of existing training images by making small changes. This helps the model learn to recognize cats and dogs in different situations.

The following techniques were used:

- **Random flip:** flips images left to right
- **Random rotation:** slightly turns images
- **Random zoom:** zooms in or out on parts of the image
- **Random contrast:** changes the brightness and lighting

These changes were applied only to the training data. The validation and test data were kept the same. Data augmentation helped the models learn better, improved accuracy, and made them perform well on new unseen images.



RESULTS

Model and size	Method	Validation Accuracy	Validation Loss	Test Accuracy	Test Loss
Model 1 Training : 1000 Test : 500 Val : 500	Without Augmentation	69%	0.582	68%	0.609
	With Augmentation	72%	0.571	73%	0.567
Model 2 Training : 1300 Test : 500 Val : 500	Without Augmentation	63%	0.646	61%	0.664
	With Augmentation	78%	0.54	79%	0.484
Model 3 Training : 3000 Test : 500 Val : 500	Without Augmentation	69%	0.602	63%	0.616
	With Augmentation	81%	0.463	79%	0.468
Pre-trained Model 1	With Augmentation	97%	0.111	99%	0.032
Pre-trained Model 2	With Augmentation	97%	0.081	98%	0.039
Pre-trained Model 3	With Augmentation	96%	0.136	98%	0.039

Best Model built from scratch and from a pre-trained model

Model and size	Method	Validation Accuracy	Validation Loss	Test Accuracy	Test Loss
Model 3 Training : 3000 Test : 500 Val : 500	With Augmentation	81%	0.463	79%	0.468
Pre-trained Model 1	With Augmentation	97%	0.111	99%	0.032

Images predicted by the Pre-trained Model 1

Images predicted by the Model built from scratch – Model 3 (with Augmentation)

Relationship between training sample size and choice of network

1. **Model 1 , Small training set (around 1,000 images)**
   The model built from scratch (with augmentation) achieved 72 % validation accuracy and 73 % test accuracy with a validation loss of 0.571 and test loss of 0.567. The pre-trained model on the same dataset reached 97 % validation accuracy and 99 % test accuracy with much lower losses (0.111 and 0.032).
   This shows that when the dataset is small, a pre-trained model performs far better than a scratch model because it already learned useful image features.

2. **Model 2 , Medium training set (around 1,300 images)**
   The model trained from scratch achieved 78 % validation accuracy and 79 % test accuracy with validation loss 0.540 and test loss 0.484. The pre-trained model achieved 97 % validation accuracy and 98 % test accuracy with validation loss 0.081 and test loss 0.039.
   The improvement in both scratch and pre-trained models shows that adding more data helps all models learn better, though transfer learning still performs best.

3. **Model 3 , Large training set (around 3,000 images)**
   The scratch model achieved 81 % validation accuracy and 79 % test accuracy with validation loss 0.463 and test loss 0.468. The pre-trained model achieved 96 % validation accuracy and 98 % test accuracy with validation loss 0.136 and test loss 0.039.
   With more training samples, even the model built from scratch shows strong generalization and higher stability.

As the training sample size increases, models trained from scratch improve steadily in accuracy and reduce loss. However, pre-trained models consistently perform better at every stage, especially when data is small. In short, small datasets work best with pre-trained networks, while larger datasets allow deeper networks to be trained successfully from scratch.

Conclusion

The best-performing model trained from scratch was Model 3, which used 3,000 training images along with 500 validation and 500 test images. This model achieved a validation accuracy of 81 %, a test accuracy of 79 %, with a validation loss of 0.463 and a test loss of 0.468. These results show that increasing the training data improved the model's ability to generalize and reduced overfitting, making it the strongest among all models built from scratch.

In comparison, the best overall performance was achieved by the pre-trained Model 1, which used MobileNetV2 with transfer learning. It reached a validation accuracy of 97 % and a test accuracy of 99 %, with a validation loss of 0.111 and a test loss of 0.032. This demonstrates the power of transfer learning, where a model pre-trained on a large dataset like ImageNet can quickly adapt to a smaller dataset and achieve exceptional accuracy.

Overall, the findings show that while increasing the training sample size improves the performance of models trained from scratch, pre-trained models consistently outperform them, especially when the available data is limited. Therefore, transfer learning provides a highly efficient approach for achieving high accuracy in image classification tasks with smaller datasets, while larger datasets make training deep networks from scratch more practical and effective.