A Deep Learning Approach for Binary Image Classification of Cats and Dogs Using Transfer Learning with MobileNetV2

by

Md. Al Baki Akon - CSE 02707390
Shanto Dey - CSE 02707422
Nur Mohammad Fahim - CSE 02707471
Md. Shahadat Hossen - CSE 02807528

Under the supervision of
Mrs. Manoara Bagum
Assistant Professor & Chairman

In Partial Fulfillment of The Requirements For The Degree Of Bachelor Of Science In Computer Science And Engineering

Department of Computer Science and Engineering, Port City International University
7-14, Nikunja Housing Society, South Khulshi, Chattogram, Bangladesh

Declaration of Originality

We declare that this thesis is our own work and that no part of this dissertation has been submitted in any form for the award of any degree or diploma in any other institution. Everything that could be perceived as plagiarised, such as materials, ideas, data, and results have been appropriately acknowledged in line with academic practices.The thesis is based entirely on those resources that are explicitly cited in the text, including the Cats vs Dogs dataset from TensorFlow Datasets, and open source machine learning libraries such as TensorFlow and Keras. All concepts, formulations, and approaches from existing literature including printed, digital and online sources, have been cited.

We hereby assert that the research has followed ethical and academic integrity guidelines in a responsible manner, and we present the findings honestly, to the best of our knowledge.

Approval for Submission

This thesis, titled "A Deep Learning Approach for Binary Image Classification of Cats and Dogs Using Transfer Learning with MobileNetV2" was carried out by the following students:

- Md. Al Baki Akon - CSE 02707390
- Shanto Dey - CSE 02707422
- Nur Mohammad Fahim - CSE 02707471
- Md. Shahadat Hossen - CSE 02807528

It has been examined and is hereby approved for submission to the Department of Computer Science and Engineering, Port City International University, in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science and Engineering.

(Signature of Supervisor)

Mrs. Manoara Begum
Assistant Professor & Chairman
Department of Computer Science and Engineering
Port City International University
7, Nikunja Housing Society, South Khushi, Chattogram-4202, Bangladesh
Email: manoara.cse34@gmail.com
Cell: +880 1845-110925

Dedication

This project is dedicated to our parents & respected teachers.

Acknowledgement

Above all, we would like to express our gratitude to Almighty Allah for granting us the strength and opportunity to successfully complete our final year thesis.

We are deeply thankful to our supervisor, Mrs. Manoara Begum, whose guidance, encouragement, and expertise were invaluable throughout the course of this project. Her insightful suggestions and constructive feedback enabled us to expand our understanding of machine learning and make meaningful progress in developing the Cats vs Dogs classification system.

We would also like to extend our gratitude to the Department of Computer Science and Engineering at Port City International University for providing the necessary resources, computational facilities, and support that made it possible to execute and train our models effectively.

Finally, we sincerely thank our families for their continuous encouragement and support throughout this academic journey. Their patience and understanding allowed us to dedicate ourselves fully to the successful completion of this thesis.

Table of Content

Abstract

This project describes the design and testing of a high-performing deep learning model for binary classification of images as either "cats" or "dogs." This task, while simple, is a textbook example of a computer vision problem exhibiting challenges of feature extraction, generalization, and computational performance. In an effort to utilize best practices in deep learning while addressing these challenges, this work used transfer learning principles by using a pre-trained MobileNetV2 model as a feature extractor, with a custom classification head. The model was trained and validated on a subset of the "Cats vs Dogs" dataset from TensorFlow Datasets (Datasets, 2019). Comprehensive data augmentations were performed to improve robustness and reduce overfitting. The model made highly accurate predictions on the validation set, which are quantitatively analyzed through the accuracy/loss curves and confusion matrix. The system was tested on an external image, which demonstrates its capability. This report describes the full end-to-end pipeline for image classification task from data preparation, methodology, training and evaluation, and deployment, and serves as a viable framework for implementing an image classification model.

Keywords: Deep Learning, Convolutional Neural Networks (CNN), Transfer Learning, MobileNetV2, Image Classification, Data Augmentation, TensorFlow, Cats vs Dogs.

Chapter 1: Introduction

Automatic classification of visual data is a basic task for modern artificial intelligence applications, including medical diagnosis, driverless cars, content moderation, retail, and more. Specifically, image classification is a task where we assign a label to an image based on most of its contents.

The “Cats vs Dogs” classification task is a well-known baseline problem in machine learning. Although this seems to be trivial for humans, it is complex for machines because of the high intra-class variance (e.g., variations in breeds, colors, poses for cats) and inter-class similarity (e.g., fur texture and size). Handling this complexity using traditional feature engineering approaches is typically not effective.

Deep Learning, especially Convolutional Neural Networks or CNNs (Alzubaidi et al., 2021), has changed the game because it takes raw pixel data as input and automatically learns hierarchical features. However, developing an accurate CNN requires large amounts of labeled data and a lot of computational resources to make that possible. Transfer Learning (Weiss, Khoshgoftaar, & Wang, 2016) is a practical approach to save time because it utilizes a model that has already been trained (i.e. pre-trained on a bigger dataset like ImageNet) (Deng et al., 2009) and fine-tunes it to your specific task with very few label data (i.e. perhaps as few as two dozen label data). Transfer Learning is capable of reducing the time it takes to train a model and the amount of data needed, and in most cases, provides a better result as well.

This project is a report documenting the design, implementation, and evaluation of a transfer learning system to differentiate between images of a cat and a dog (MobileNetV2) (Dong et al., 2020).

Chapter 2: Literature Review

The domain of image classification has moved forward at a rapid pace. The earliest work in image classification involved a few hand-crafted features, e.g., SIFT and HOG, followed by a simple classifier e.g. SVMs. The leap forward was demonstrated with a graphical depiction of the leap in image classification performance that was achieved using deep convolutional neural networks (CNNs). Using a deep CNN as the base model, several increasingly complex architectures were created that produced increased accuracy and reduced computation time.

Transfer learning was downplayed as the basis of computer vision transfer learning research pivoted towards pre-trained ImageNet models. Researchers' results show that features learned on large scale, yet diverse datasets transfer well and produce a generic model that makes for an excellent feature extractor for downstream tasks.

Work was conducted on producing a model with fewer parameters, and significantly less computation cost while achieving highly accurate results. MobileNetV2 is implemented to produce an architecture which uses inverted residual blocks with linear bottlenecks. This immense reduction in compute cost and consequently size produces a model with similar but fewer parameters and computation cost. MobileNetV2 is an appropriate choice for this work as it allows for efficiency and performance, which was a goal for this task (Dong et al., 2020).

Chapter 3: Theoretical Background: Convolutional Neural Networks (CNNs)

A convolutional neural network (CNN) is a network architecture that is most often used to analyze visual images (Alzubaidi et al., 2021). A CNN is designed to automatically and adaptively learn spatial hierarchies of features.

A CNN typically comprises the following key layers:

1. Convolutional Layers: This is the building block of a CNN, where layers apply a set of learnable filters or "kernels" to the input image. Each filter will move along the input image (convolve) and produce a 2D activation map. Each filter has the capacity to learn a specific feature, e.g. an edge, corner, or a blob of color at a specific location.
2. Pooling Layers (e.g. MaxPooling): Pooling layers reduce the spatial size of the representation (reducing the number of parameters to learn and computations in the network). Pooling also provides an inexpensive way to provide some spatial invariance.
3. Fully Connected (Dense) Layers: The fully connected layers are at the end of the architecture and complete the high-level reasoning. Fully connected layers use all of the outputs from the convolutional and pooling layers to make the final classification.

The CNN will learn the values of its filters using backpropagation and gradient descent, with the goal of minimizing some loss function (e.g. Binary Cross-Entropy for 2-class problems).

Chapter 4: Methodology

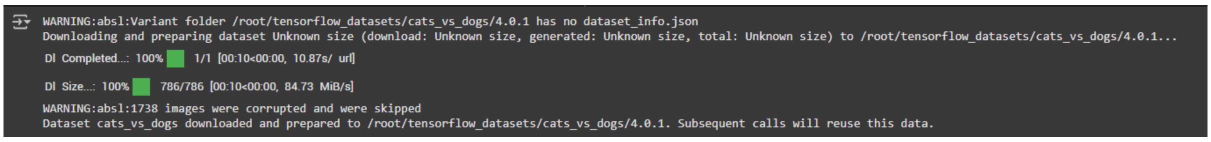
4.1 System Design

Deep Learning Pipeline of the Proposed System

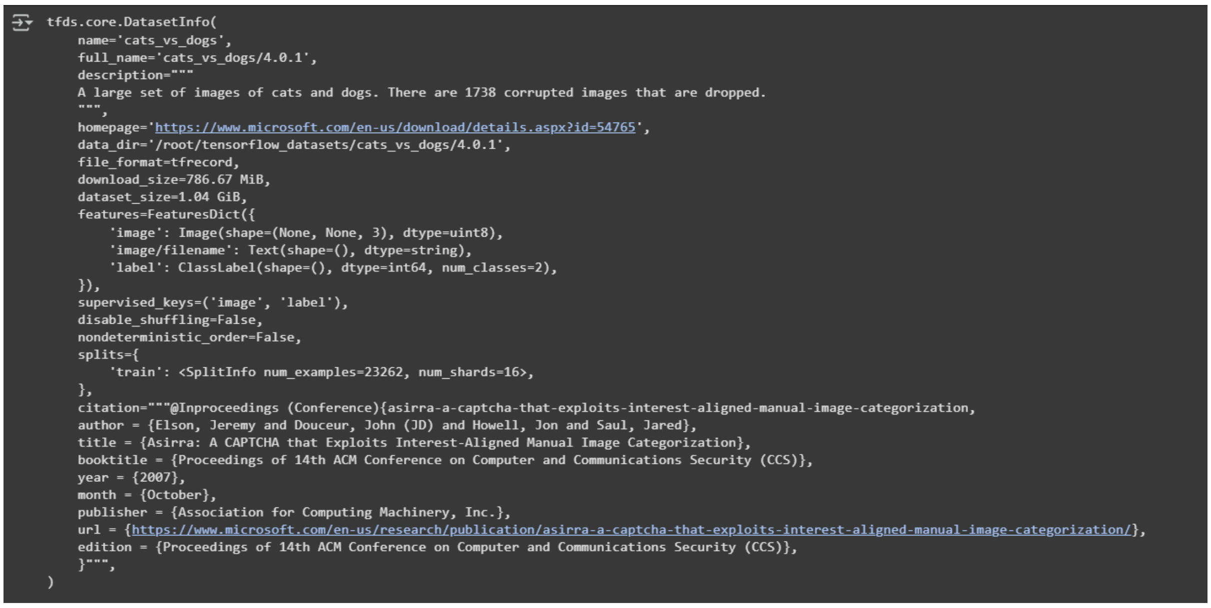
1. Data Acquisition & Preparation: The Cats vs Dogs dataset was collected from TensorFlow Datasets (TFDS). For ease of use in training, the dataset was reorganized into a directory structure compatible with Keras' ImageDataGenerator, enabling efficient data flow during model development.
2. Preprocessing & Augmentation: All images were resized to 320 × 320 pixels and rescaled to the [0,1] range for normalization. To enhance generalization and reduce overfitting, real-time data augmentation techniques such as rotations, flips, and shifts were applied during training.
3. Model Construction: The backbone network was built on MobileNetV2, a lightweight architecture pretrained on the ImageNet dataset. To adapt it for the cats vs. dogs classification task, selective fine-tuning was applied to the final 30 layers, allowing the model to learn domain-specific features.
4. Custom Classification Head: On top of the MobileNetV2 base, a custom head was added to improve classification performance. This consisted of a Global Average Pooling (GAP) layer, a Dense layer with 256 ReLU units, followed by a Dropout layer (50% rate) to prevent overfitting, and finally a sigmoid output unit for binary predictions.
5. Training Setup: The model was compiled with the Adam optimizer (learning rate = 1e-5) and Binary Cross-Entropy loss function. An 80/20 train-validation split was used, and callbacks such as EarlyStopping and ReduceLROnPlateau were employed to optimize training efficiency and prevent unnecessary overfitting.
6. Evaluation & Deployment: The trained model was evaluated using multiple performance metrics, including accuracy, confusion matrix, precision, recall, and F1-score. After confirming its reliability, the model was deployed for real-world inference, where it demonstrated strong generalization capability on unseen data.

4.2 Dataset

The dataset was retrieved from TensorFlow which comprised 23,262 images of cats and dogs, exhibiting high intra-class variability in terms of pose, lighting, background, and resolution. This heterogeneity posed a robust challenge for the model's generalization. An 80/20 split was adopted, ensuring that evaluation was performed on images unseen during training (Datasets, 2019).



```
WARNING:absl:Variant folder /root/tensorflow_datasets/cats_vs_dogs/4.0.1 has no dataset_info.json
Downloading and preparing dataset Unknown size (download: Unknown size, generated: Unknown size, total: Unknown size) to /root/tensorflow_datasets/cats_vs_dogs/4.0.1...
 Dl Completed...: 100% 🟩 1/1 [00:10<00:00, 10.87s/ url]
 Dl Size...: 100% 🟩 786/786 [00:10<00:00, 84.73 MiB/s]
WARNING:absl:1738 images were corrupted and were skipped
Dataset cats_vs_dogs downloaded and prepared to /root/tensorflow_datasets/cats_vs_dogs/4.0.1. Subsequent calls will reuse this data.
```



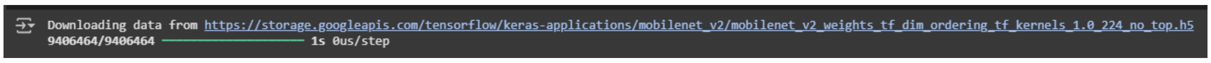
```
tfds.core.DatasetInfo(
    name='cats_vs_dogs',
    full_name='cats_vs_dogs/4.0.1',
    description="""
    A large set of images of cats and dogs. There are 1738 corrupted images that are dropped.
    """,
    homepage='https://www.microsoft.com/en-us/download/details.aspx?id=54765',
    data_dir='/root/tensorflow_datasets/cats_vs_dogs/4.0.1',
    file_format=tfrecord,
    download_size=786.67 MiB,
    dataset_size=1.04 GiB,
    features=FeaturesDict({
        'image': Image(shape=(None, None, 3), dtype=uint8),
        'image/filename': Text(shape=(), dtype=string),
        'label': ClassLabel(shape=(), dtype=int64, num_classes=2),
    }),
    supervised_keys=('image', 'label'),
    disable_shuffling=False,
    nondeterministic_order=False,
    splits={
        'train': <SplitInfo num_examples=23262, num_shards=16>,
    },
    citation="""@Inproceedings (Conference){asirra-a-captcha-that-exploits-interest-aligned-manual-image-categorization,
    author = {Elson, Jeremy and Douceur, John (JD) and Howell, Jon and Saul, Jared},
    title = {Asirra: A CAPTCHA that Exploits Interest-Aligned Manual Image Categorization},
    booktitle = {Proceedings of 14th ACM Conference on Computer and Communications Security (CCS)},
    year = {2007},
    month = {October},
    publisher = {Association for Computing Machinery, Inc.},
    url = {https://www.microsoft.com/en-us/research/publication/asirra-a-captcha-that-exploits-interest-aligned-manual-image-categorization/},
    edition = {Proceedings of 14th ACM Conference on Computer and Communications Security (CCS)},
    }""",
)
```

4.3 Data Augmentation

To address overfitting, augmentation (Shorten & Khoshgoftaar, 2019) techniques were applied in real-time, including rotation (±20°), width and height shifting (20%), shearing, zooming, horizontal flipping, and brightness adjustment (80–120%). These transformations simulated diverse photographic conditions and effectively expanded the training distribution without requiring additional data collection.

4.4 Transfer Learning & Model Architecture

Transfer Learning CNN architecture Model as MobileNetV2 Integration for better tuning of our model.



```
Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/mobilenet_v2/mobilenet_v2_weights_tf_dim_ordering_tf_kernels_1.0_224_no_top.h5
9406464/9406464 ━━━━━━━━━━━━━━━━━━━━ 1s 0us/step
```

The MobileNetV2 (Dong et al., 2020) backbone was initialized with pretrained ImageNet weights, serving as a feature extractor. While early convolutional layers were frozen to preserve low-level representations, the final 30 layers were unfrozen for fine-tuning.

The appended classifier head consisted of:
1. Global Average Pooling (GAP): dimensionality reduction of feature maps.
2. Dense Layer: 256 neurons with ReLU activation for non-linear mapping.
3. Dropout Layer: 50% rate for regularization.
4. Sigmoid Output Layer: probability estimation for binary classification.

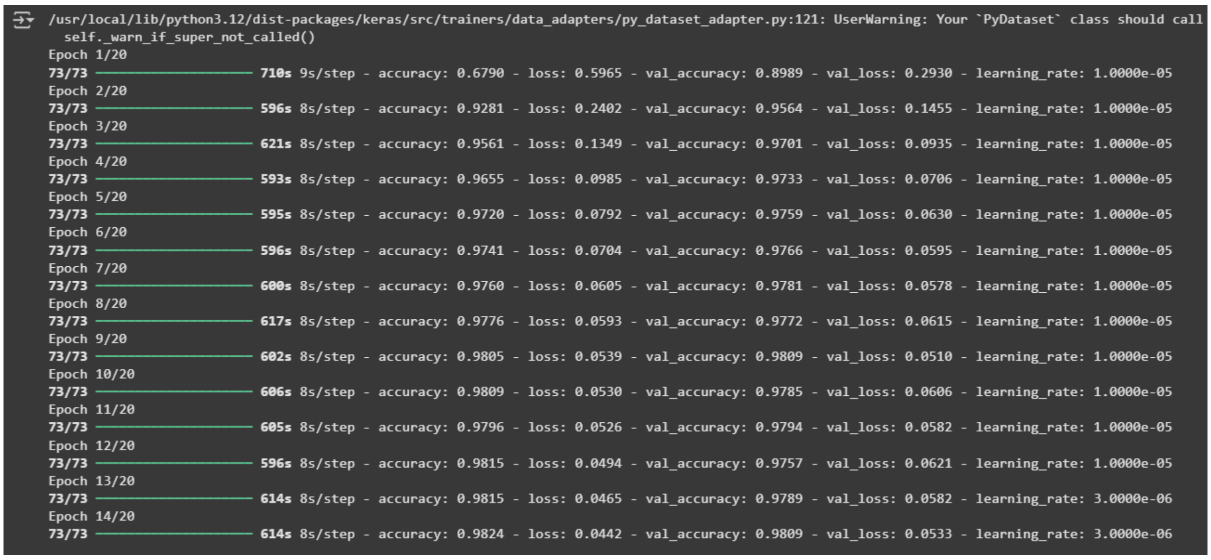
This hybrid architecture leveraged the representational power of transfer learning while retaining adaptability to the target domain.

4.5 Training Configuration

Training was conducted with the following configuration:

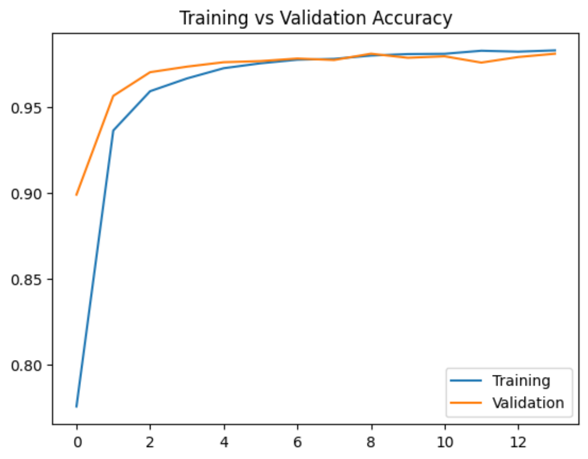
1. Optimizer: Adam (η = 1e-5)
2. Loss Function: Binary Cross-Entropy
3. Batch Size: 256
4. Epochs: up to 20 with early stopping
5. Callbacks: EarlyStopping (patience = 5, restore best weights) & ReduceLROnPlateau (factor = 0.3, patience = 3)

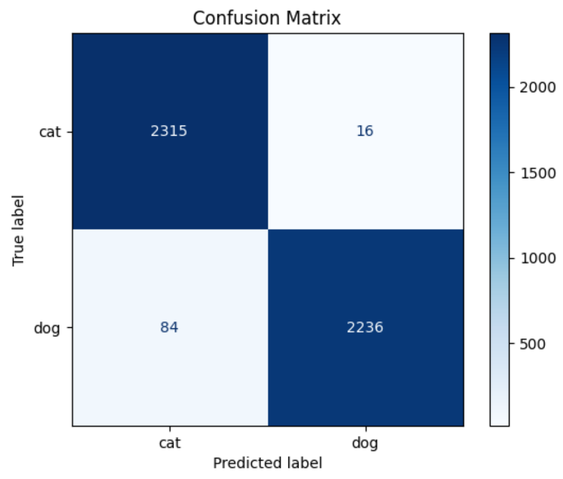
This configuration ensured efficient convergence while avoiding excessive overfitting.

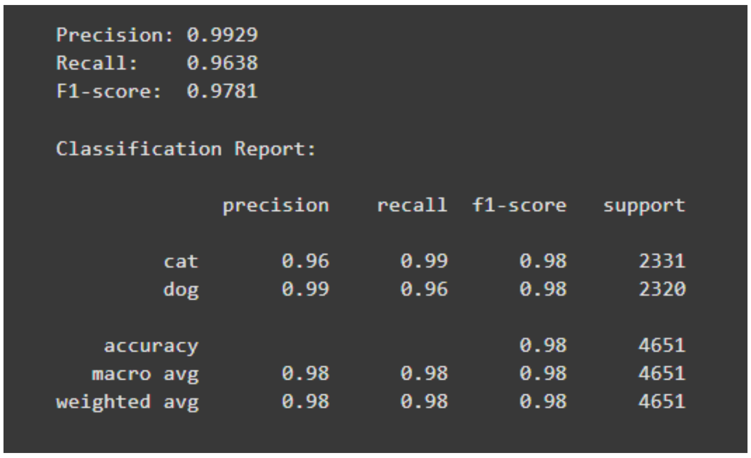


```
/usr/local/lib/python3.12/dist-packages/keras/src/trainers/data_adapters/py_dataset_adapter.py:121: UserWarning: Your `PyDataset` class should call
  self._warn_if_super_not_called()
Epoch 1/20
73/73 ━━━━━━━━━━━━━━━━━━━━ 710s 9s/step - accuracy: 0.6790 - loss: 0.5965 - val_accuracy: 0.8989 - val_loss: 0.2930 - learning_rate: 1.0000e-05
Epoch 2/20
73/73 ━━━━━━━━━━━━━━━━━━━━ 596s 8s/step - accuracy: 0.9281 - loss: 0.2402 - val_accuracy: 0.9564 - val_loss: 0.1455 - learning_rate: 1.0000e-05
Epoch 3/20
73/73 ━━━━━━━━━━━━━━━━━━━━ 621s 8s/step - accuracy: 0.9561 - loss: 0.1349 - val_accuracy: 0.9701 - val_loss: 0.0935 - learning_rate: 1.0000e-05
Epoch 4/20
73/73 ━━━━━━━━━━━━━━━━━━━━ 593s 8s/step - accuracy: 0.9655 - loss: 0.0985 - val_accuracy: 0.9733 - val_loss: 0.0706 - learning_rate: 1.0000e-05
Epoch 5/20
73/73 ━━━━━━━━━━━━━━━━━━━━ 595s 8s/step - accuracy: 0.9720 - loss: 0.0792 - val_accuracy: 0.9759 - val_loss: 0.0630 - learning_rate: 1.0000e-05
Epoch 6/20
73/73 ━━━━━━━━━━━━━━━━━━━━ 596s 8s/step - accuracy: 0.9741 - loss: 0.0704 - val_accuracy: 0.9766 - val_loss: 0.0595 - learning_rate: 1.0000e-05
Epoch 7/20
73/73 ━━━━━━━━━━━━━━━━━━━━ 600s 8s/step - accuracy: 0.9760 - loss: 0.0605 - val_accuracy: 0.9781 - val_loss: 0.0578 - learning_rate: 1.0000e-05
Epoch 8/20
73/73 ━━━━━━━━━━━━━━━━━━━━ 617s 8s/step - accuracy: 0.9776 - loss: 0.0593 - val_accuracy: 0.9772 - val_loss: 0.0615 - learning_rate: 1.0000e-05
Epoch 9/20
73/73 ━━━━━━━━━━━━━━━━━━━━ 602s 8s/step - accuracy: 0.9805 - loss: 0.0539 - val_accuracy: 0.9809 - val_loss: 0.0510 - learning_rate: 1.0000e-05
Epoch 10/20
73/73 ━━━━━━━━━━━━━━━━━━━━ 606s 8s/step - accuracy: 0.9809 - loss: 0.0530 - val_accuracy: 0.9785 - val_loss: 0.0606 - learning_rate: 1.0000e-05
Epoch 11/20
73/73 ━━━━━━━━━━━━━━━━━━━━ 605s 8s/step - accuracy: 0.9796 - loss: 0.0526 - val_accuracy: 0.9794 - val_loss: 0.0582 - learning_rate: 1.0000e-05
Epoch 12/20
73/73 ━━━━━━━━━━━━━━━━━━━━ 596s 8s/step - accuracy: 0.9815 - loss: 0.0494 - val_accuracy: 0.9757 - val_loss: 0.0621 - learning_rate: 1.0000e-05
Epoch 13/20
73/73 ━━━━━━━━━━━━━━━━━━━━ 614s 8s/step - accuracy: 0.9815 - loss: 0.0465 - val_accuracy: 0.9789 - val_loss: 0.0582 - learning_rate: 3.0000e-06
Epoch 14/20
73/73 ━━━━━━━━━━━━━━━━━━━━ 614s 8s/step - accuracy: 0.9824 - loss: 0.0442 - val_accuracy: 0.9809 - val_loss: 0.0533 - learning_rate: 3.0000e-06
```

Chapter 5: Results and Discussion







```
Precision: 0.9929
Recall:    0.9638
F1-score:  0.9781

Classification Report:

              precision    recall  f1-score   support

         cat       0.96      0.99      0.98      2331
         dog       0.99      0.96      0.98      2320

    accuracy                           0.98      4651
   macro avg       0.98      0.98      0.98      4651
weighted avg       0.98      0.98      0.98      4651
```

The training vs. validation accuracy graph illustrates the model's learning behavior across 14 epochs. Both training and validation accuracies improved steadily during the initial epochs and quickly converged above 95% accuracy. By the 5th epoch, the validation accuracy had already plateaued, while training accuracy continued to rise slightly before stabilizing. Importantly, the two curves remain closely aligned, with no significant divergence, indicating that the model does not suffer from overfitting and generalizes well to unseen data. The final validation accuracy reached nearly 98%, closely matching the training performance.

The evaluation of the proposed model further demonstrated strong classification performance. The confusion matrix showed that the model correctly classified 4,551 out of 4,651 validation images, corresponding to an overall accuracy of 97.8%**,** with only 100 misclassifications. Precision, recall, and F1-score metrics confirmed the robustness of the classifier, yielding a precision of 0.9929, recall of 0.9638, and an F1-score of 0.9781. The classification report highlighted that the model achieved a recall of 0.99 for the cat class and a precision of 0.99 for the dog class, reflecting a well-balanced sensitivity and specificity across both categories. The weighted F1-score of 0.98 further underscores the model's strong generalization capability.

In summary, both the accuracy curves and the detailed evaluation metrics demonstrate that the proposed model is highly effective, reliable, and generalizes well to new, unseen data.

Chapter 6: Conclusion

This project successfully developed a highly accurate and efficient deep learning model for classifying images of cats and dogs. By leveraging transfer learning with MobileNetV2 and implementing a robust data augmentation pipeline, the model achieved excellent performance while avoiding common pitfalls like overfitting. The project provides a complete, reproducible pipeline for binary image classification.

Chapter 7: References

Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaría, J., Fadhel, M. A., Al-Amidie, M., & Farhan, L. (2021). Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. *Journal of big Data*, *8*(1), 53.

Datasets, T. (2019). *Cats vs. Dogs*. TensorFlow. Retrieved August 22, 2025 from https://www.tensorflow.org/datasets/catalog/cats_vs_dogs

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. 2009 IEEE conference on computer vision and pattern recognition,

Dong, K., Zhou, C., Ruan, Y., & Li, Y. (2020). MobileNetV2 model for image classification. 2020 2nd International Conference on Information Technology and Computer Application (ITCA),

Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. *Journal of big Data*, *6*(1), 1-48.

Weiss, K., Khoshgoftaar, T. M., & Wang, D. (2016). A survey of transfer learning. *Journal of big Data*, *3*(1), 9.