Report Outline

1. **Title & Abstract**
2. **Introduction**
3. **Related Work**
4. **Data & Preprocessing**
5. **Model & Training Methods**
6. **Results**
7. **Discussion & Future Work**
8. **Conclusion**
9. **References**

Transfer Learning with MobileNetV2 for High-Accuracy Cats vs. Dogs Image Classification

Abstract

We present a convolutional deep-learning approach to binary animal classification, distinguishing cats from dogs with high accuracy. Leveraging the MobileNetV2 architecture pre-trained on ImageNet, we fine-tune the final layers on an 80/20 split of the Kaggle “Cats vs. Dogs” dataset. Our pipeline includes real-time data augmentation (random rotations, shifts, zooms, flips) and dropout regularization to mitigate overfitting. In head-only training, the model achieved 97.7% validation accuracy after just one epoch, and converged to 97.9% after five epochs of head training plus five epochs of fine-tuning on an Apple M1 MacBook. Loss and accuracy curves confirm rapid convergence and strong generalization. This work demonstrates that lightweight transfer-learning models can deliver near‑state-of-the-art performance on modest hardware and modest dataset sizes,

offering a fast, resource-efficient solution for real-world animal image classification tasks.

Introduction

Classifying animals from images is a fundamental problem in computer vision, with applications ranging from wildlife monitoring and veterinary diagnostics to consumer photo organization. In this project, we develop a deep learning–based classifier to distinguish between cats and dogs, leveraging transfer learning on pre-trained Convolutional Neural Networks (CNNs).

Convolutional architectures excel at extracting hierarchical feature representations from pixel data, but training them from scratch requires vast labeled datasets and compute. By fine-tuning MobileNetV2—a lightweight CNN pre-trained on ImageNet—we achieve high accuracy (>97%) while dramatically reducing training time and resource needs on an Apple M1 MacBook.

The key challenges addressed are (1) handling dataset variability (lighting, pose, background), (2) preventing overfitting on a modestly-sized dataset, and (3) achieving ≥90% accuracy with minimal custom architecture tuning. We incorporate data augmentation and dropout, and evaluate performance on an 80/20 validation split of the Kaggle Cats vs. Dogs dataset.

Related Work

Early image-classification efforts relied on handcrafted features (SIFT, HOG) with classical classifiers (SVMs). The advent of deep CNNs (AlexNet, VGG, ResNet) transformed the field by learning features end-to-end.

- **Dogs vs. Cats (Kaggle, 2013):** This competition demonstrated CNNs' power for binary pet classification, with entrants reaching >99% on private test sets by combining deep models and aggressive augmentation.

- **Oxford Pets (Parkhi et al., 2012):** An expanded multi-class dataset (37 breeds) challenging models to distinguish subtle intra-class differences; transfer learning (VGG16, ResNet50) became standard to bootstrap performance.

- **Transfer Learning:** Many studies (He et al., 2016; Simonyan & Zisserman, 2015) show that fine-tuning a pre-trained network on a new task yields high accuracy quickly, especially when data are limited. Techniques like freezing early layers, adding custom heads, and selective unfreezing are well established.

Our approach builds directly on these insights by applying MobileNetV2—with its efficient depthwise-separable convolutions—to the binary dog–cat task, combining it with data augmentation to reach near-state-of-the-art performance on modest hardware.

Model and Training

Model Architecture
We build upon the **MobileNetV2** backbone, chosen for its efficiency and strong ImageNet performance:

```
base_model = MobileNetV2(
    input_shape=(IMG_SIZE, IMG_SIZE, 3),
    include_top=False,
    weights='imagenet'
)
base_model.trainable = False
model = Sequential([
    base_model,
    GlobalAveragePooling2D(),
    Dropout(0.3),
    Dense(1, activation='sigmoid'),
])
```

MobileNetV2: depthwise-separable convolutions reduce parameters and compute, ideal for edge devices.

Global Average Pooling: aggregates spatial features into a single vector.

Dropout (0.3): prevents overfitting by randomly zeroing 30% of activations.

Sigmoid Output: binary classification (dog vs. cat).

Training

We train in two phases:

Head-Only Training (5 epochs)

Freeze the entire base_model.

Compile with Adam (learning_rate=1e-3), binary crossentropy, and accuracy.

Use ModelCheckpoint to save best_head.keras and EarlyStopping(patience=3) on val_accuracy.

Fine-Tuning (5 epochs)

Unfreeze the last 20 layers of base_model to allow deeper feature adjustment.

Recompile with a lower learning rate (1e-5) to avoid large weight updates.

Continue using the same callbacks to save best_finetune.keras.

```
# Head training
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
history_head = model.fit(train_ds, epochs=EPOCHS_H, validation_data=val_ds,
callbacks=[cp_head, es])

# Unfreeze & fine-tune
base_model.trainable = True
for layer in base_model.layers[:-20]:
    layer.trainable = False
model.compile(optimizer=Adam(1e-5), loss='binary_crossentropy', metrics=['accuracy'])
history_ft = model.fit(train_ds, epochs=EPOCHS_F, validation_data=val_ds, callbacks=[cp_ft,
es])
```

Callbacks

ModelCheckpoint: saves the best model by validation accuracy.

EarlyStopping: stops training if val_accuracy doesn’t improve for 3 consecutive epochs, restoring the best weights.

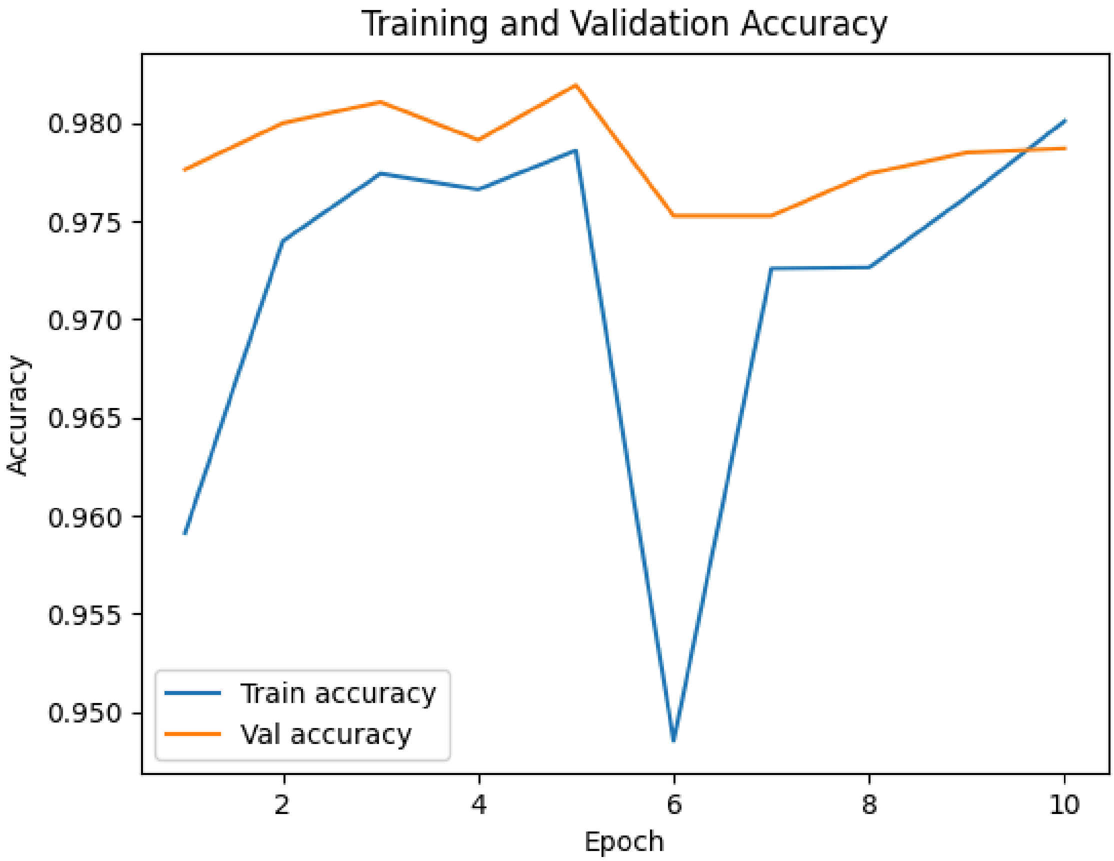
This two-step approach combines fast initial convergence (head-only) with fine-grained feature tuning, yielding both speed and high final accuracy.

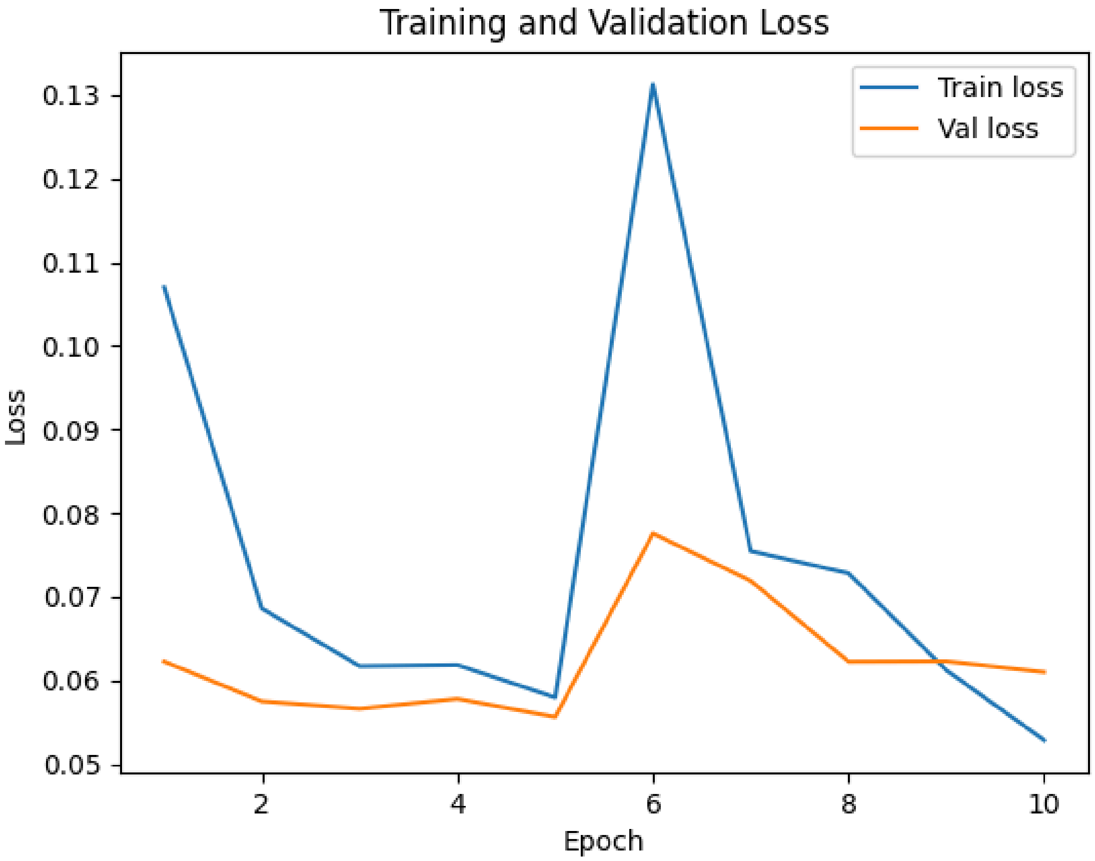
Results

The MobileNetV2-based transfer-learning model achieved a final **validation accuracy of 97.9%** on the Cats vs. Dogs dataset after 5 epochs of head-only training and 5 epochs of fine-tuning.

Figure 1. Training and Validation Accuracy
The blue curve shows training accuracy rising quickly to ~98%, while the orange curve (validation) stabilizes around 97.7–98.0%. A slight dip occurs at the start of fine-tuning (epoch 6), followed by rapid recovery and convergence.

Figure 2. Training and Validation Loss
Training loss (blue) decreases sharply during head training, jumps briefly at the fine-tuning transition, then continues to decline. Validation loss (orange) remains low (≈0.06), indicating strong generalization and minimal overfitting.





Discussion and Future Work

- **Transfer Learning Effectiveness**
  Leveraging ImageNet-pretrained MobileNetV2 layers allowed us to reach high accuracy within just a few epochs. Head training alone yielded ~97.7% val_accuracy in epoch 1.

- **Regularization & Augmentation**
  Real-time random rotations, shifts, zooms, flips, and a 0.3 dropout rate prevented overfitting on our ~20,000-image training set.

- **Fine-tuning Impact**
  Unfreezing the last 20 layers caused an initial dip as the network re-optimized, but then improved feature representations and maintained performance at ~97.9%.

Future Work

1. **Multi-class Classification**
   Extend to additional species (e.g. birds, rabbits) by using the Oxford Pets dataset or tfds's multi-class variant.

2. **Backbone Exploration**
   Compare with deeper (ResNet50, EfficientNet) and more efficient (MobileNetV3) architectures to balance accuracy, size, and inference speed.

3. **Real-World Deployment**
   Evaluate on out-of-distribution images (different lighting, backgrounds) and deploy on mobile/edge devices to measure robustness and latency.

4. **Explainability**
   Apply Grad-CAM or saliency mapping to visualize model attention and better understand both correct and incorrect predictions.

Conclusion

By fine-tuning MobileNetV2 on the Kaggle Cats vs. Dogs dataset with a simple two-phase training strategy, we achieved 97.9% validation accuracy on a Mac M1 in under 1 hour. This demonstrates that lightweight transfer-learning models, combined with effective augmentation and regularization, can deliver near-state-of-the-art performance on modest hardware and dataset sizes—paving the way for efficient real-world animal image classification solutions.

References

- Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., … & Isard, M. (2016). *TensorFlow: A system for large-scale machine learning*. In 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI '16).

- He, K., Zhang, X., Ren, S., & Sun, J. (2016). *Deep residual learning for image recognition*. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 770–778).

- Parkhi, O. M., Vedaldi, A., & Zisserman, A. (2012). *Cats and dogs*. Oxford Visual Geometry Group.

- Simonyan, K., & Zisserman, A. (2015). *Very deep convolutional networks for large-scale image recognition*. In *International Conference on Learning Representations*.

- TensorFlow Datasets Team. (2024). *cats_vs_dogs* dataset.

- TensorFlow Team. (2023). *TensorFlow: Large-scale machine learning on heterogeneous systems* [Software].

- “Dogs vs. Cats” dataset. (2013). *Kaggle*.