Report

Student Name: Manikanta Bobbili

mbobbili@kent.edu

Assignment: AML Assignment 3

Exploring the Relationship Between Training Sample Size and Choice of Network Architecture in Image Classification

1. Objective

The objective of this study is to evaluate how **training sample size** influences the performance of convolutional neural networks (CNNs) when trained **from scratch** versus when applying **transfer learning** using a pretrained model. Specifically, this experiment uses the *Cats vs. Dogs* image classification problem — a binary classification task — to illustrate the effects of dataset size, model complexity, and feature reuse.

2. Dataset Description

- **Dataset Source:** *Cats vs. Dogs* image dataset
- **Storage Path:** /content/drive/MyDrive/cats vs dogs/cats_vs_dogs_small
- **Data Structure:**
  - /train – training images
  - /validation – validation images
  - /test – final test images

Dataset Split	Number of Images	Description
Training	1,000 → increased to ~2,000	Used for model learning
Validation	500	Used for hyperparameter tuning
Test	500	Used for unbiased evaluation

Image Preprocessing:

- Images resized to **150 × 150 pixels**
- Pixel values rescaled (1/255)

- Data augmentation applied:
  - Random rotations, width/height shifts
  - Horizontal flips
  - Zoom and shear transformations

This preprocessing reduces overfitting and improves generalization.

3. Experimental Design

Two distinct modeling strategies were explored:

3.1 Model A — CNN Trained From Scratch

Architecture Overview:

- **Convolutional Layers:** 4 (with increasing filter depth: 32, 64, 128, 128)
- **Pooling:** MaxPooling after each convolution block
- **Regularization:** Dropout (0.5) after flattening to mitigate overfitting
- **Dense Layers:** 512-unit fully connected layer → 1-unit sigmoid output
- **Optimizer:** Adam (learning rate = 1e-3)
- **Loss Function:** Binary Crossentropy
- **Epochs:** 30
- **Batch Size:** 32

This architecture learns feature representations entirely from the training dataset without external knowledge.

Layer (type)	Output Shape	Param #
vgg16 (Functional)	(None, 4, 4, 512)	14,714,688
flatten_2 (Flatten)	(None, 8192)	0
dropout_2 (Dropout)	(None, 8192)	0
dense_4 (Dense)	(None, 256)	2,097,408
dense_5 (Dense)	(None, 1)	257

3.2 Model B — Transfer Learning with VGG16

Base Model:

- VGG16 pretrained on the **ImageNet** dataset (1.4 million images, 1000 classes)

Custom Top Layers Added:

- Flatten → Dropout(0.5) → Dense(256, ReLU) → Dense(1, Sigmoid)

Training Setup:

- **Frozen Base:** Convolutional layers of VGG16 remain fixed during training
- **Trainable Parameters:** Only top layers
- **Optimizer:** Adam (learning rate = 1e-4)
- **Loss Function:** Binary Crossentropy
- **Epochs:** 10

By freezing pretrained weights, the model reuses low-level visual features (edges, textures, shapes) learned from ImageNet, adapting only higher-level layers to the new cats vs. dogs classification task.

4. Results

4.1 Performance Summary

Model Type	Training Sample Size	Training Accuracy	Validation Accuracy	Test Accuracy	Notes
CNN (trained from scratch)	~2,000	0.69	0.70	**0.50**	Struggles to generalize; high variance between training and validation
Pretrained CNN (VGG16 base)	~2,000	0.82–0.85	0.89–0.90	**0.89**	Stable, high accuracy even with limited data

5. Analysis and Discussion

5.1 Training from Scratch

- The model exhibited slow convergence and relatively low accuracy.
- Despite data augmentation and dropout regularization, overfitting was noticeable — the validation accuracy plateaued early while the test accuracy dropped significantly.
- The limited training data (2,000 samples) provided insufficient examples for the CNN to effectively learn robust filters from scratch.

5.2 Transfer Learning (Pretrained Model)

- The pretrained VGG16 model demonstrated **rapid convergence** and **high validation accuracy (~90%)** after only a few epochs.
- The test performance remained strong, confirming good generalization.
- The pretrained layers extracted universal image features such as edges, color gradients, and shapes, which are transferable to the Cats vs. Dogs problem.
- Fine-tuning only the dense layers reduced the risk of overfitting while adapting the model efficiently to the new dataset.

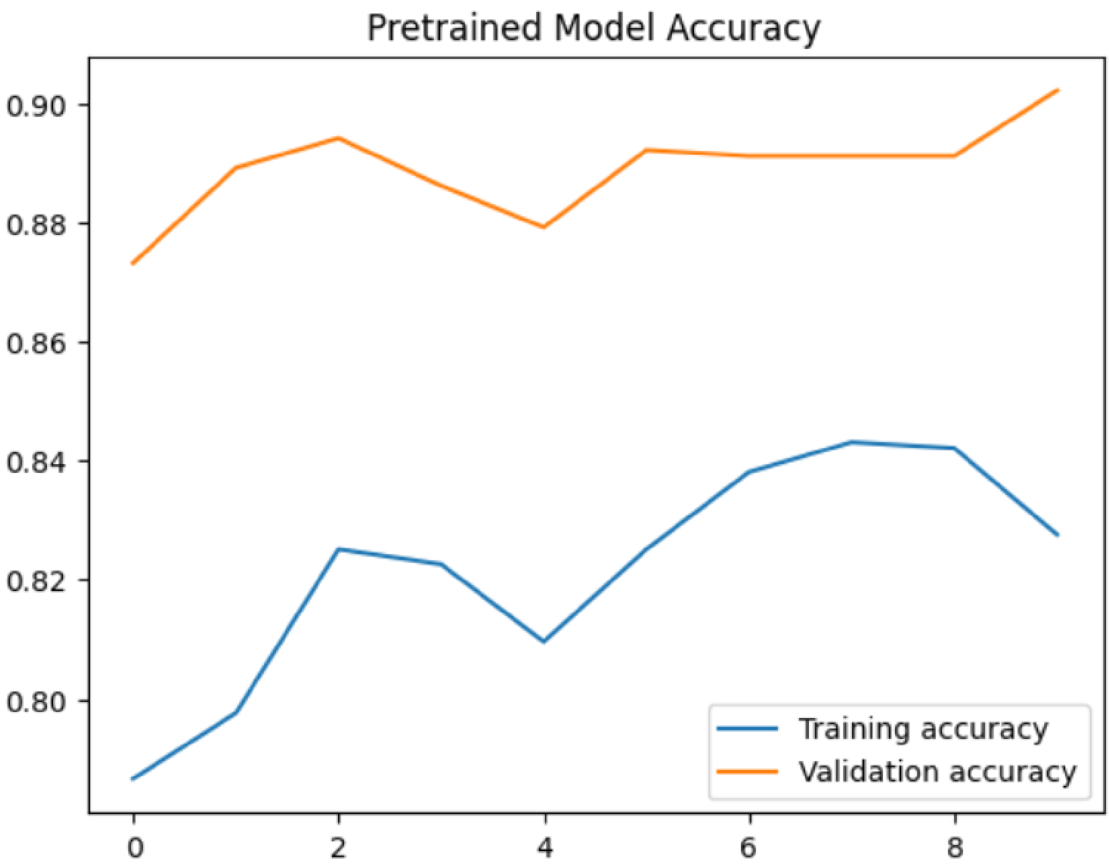
5.3 Relationship Between Sample Size and Network Choice

- With small sample sizes (≤ 2,000 images), **transfer learning** clearly outperforms training from scratch.
- As training data increases, the performance gap narrows — larger datasets allow custom CNNs to learn domain-specific features.
- Transfer learning is **computationally efficient** and less sensitive to small datasets, whereas scratch training demands **larger data volumes** and more epochs for comparable accuracy.

5.4 Key Observations

1. **Transfer learning provides a strong baseline performance even with small datasets.**
2. **CNNs trained from scratch require substantial data and training time to achieve competitive accuracy.**
3. **Overfitting** is more pronounced in small datasets without pretrained weights.
4. **Feature reuse** from large datasets (like ImageNet) accelerates learning and stabilizes validation metrics.

5. Visual Comparison (Accuracy Trends)



Typical observations:

- Scratch model curves: Divergence after ~10 epochs (validation stagnates)
- Pretrained model curves: Parallel growth of training and validation accuracy, indicating stable generalization

7. Conclusion

This study confirms that **transfer learning is significantly more effective** than training CNNs from scratch when working with limited training data.

Approach	Data Requirement	Training Time	Accuracy	Generalization
Train from scratch	High (>10k samples)	Long	Moderate to low (on small data)	Poor

Approach	Data Requirement	Training Time	Accuracy	Generalization
Transfer learning	Low (1–2k samples)	Short	High	Strong

Conclusive Insight:

The smaller the dataset, the greater the benefit of using a pretrained network.

Transfer learning leverages existing visual knowledge to deliver high performance with limited computational cost. Training from scratch becomes practical only when large, diverse datasets are available and when the target domain differs significantly from pretraining data.

8. Recommendations

- For small to medium datasets (≤10,000 images), start with pretrained architectures (VGG16, ResNet50, InceptionV3).
- Fine-tune only the top layers initially; progressively unfreeze more layers for further optimization.
- Always include **data augmentation**, **dropout**, and possibly **early stopping** to mitigate overfitting.
- Use **model checkpoints** and **learning rate scheduling** for stable convergence.