ASSIGNMENT-2 Convolution Networks

BA-64061-001

Advanced Machine Learning

Name: Sai Shripad Achutuni

E-mail ID: sachutun@kent.edu

Student ID: 811363462

ASSIGNMENT-2 SUMMARY REPORT: CONVOLUTION NETWORKS

Impact of Training Sample Size on Convolutional Neural Network Performance:

A Study Using “Cats vs Dogs Dataset”

GOAL:

Using the Cats vs. Dogs dataset, this study investigates the use of Convolutional Neural Networks (CNNs) for picture categorization tasks. Analyzing the impact of training sample size on model performance while training a network from scratch as opposed to utilizing a pretrained network (transfer learning) is the main goal. The experiment also emphasizes methods like data augmentation, dropout regularization, and early halting that minimize overfitting and enhance model generalization.

METHODOLOGY:

Dataset

- Dataset: “Cats vs Dogs”
- Original dataset split into:
    - Training: Initially 1,000 images
    - Validation: 500 images
    - Test: 500 images

In order to assess how sample size affects model performance, the dataset was further extended to bigger training sets (1,500 and 2,000 samples).

Approach A: Training the Model from Scratch

1. Architecture:

Sequential CNN with 3 convolutional blocks:

a) Conv2D (32, 64, 128 filters) with ReLU activation

b) MaxPooling after each convolution layer

c) Flatten + Dense (512) + Dropout (0.5) + Output layer (sigmoid)

2. Optimizer: Adam (lr = 1e-4)

3. Loss Function: Binary Crossentropy

4. Regularization: Data augmentation (rotation, width/height shift, zoom, horizontal flip)

5. Metrics: Accuracy on validation/test datasets

Objective: Use different sample sizes to train the model and monitor accuracy gains.

Approach B: Using the Pretrained Model

1. Base Model: VGG16 pretrained on ImageNet (top layers removed)

2. Layers Added: Flatten + Dense(256, ReLU) + Dropout(0.5) + Dense(1, Sigmoid)

3. Trainable Layers: Only top classifier layers (initially frozen base model)

4. Fine-Tuning: Gradual unfreezing of top VGG16 layers for optimization

5. Optimizer and Loss: Same as above

Objective: Evaluate performance using the same dataset sizes as the scratch model.

SAMPLE IMAGES:

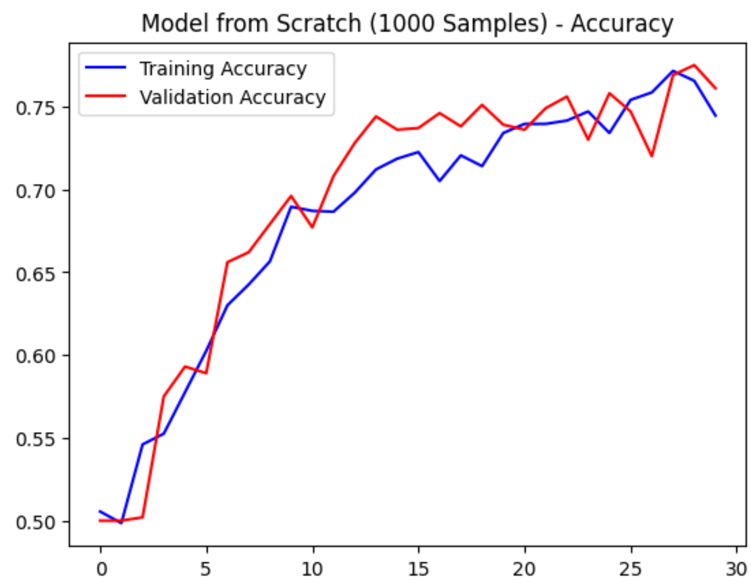
EXPERIMENT'S SUMMARY AND THEIR OUTCOMES:

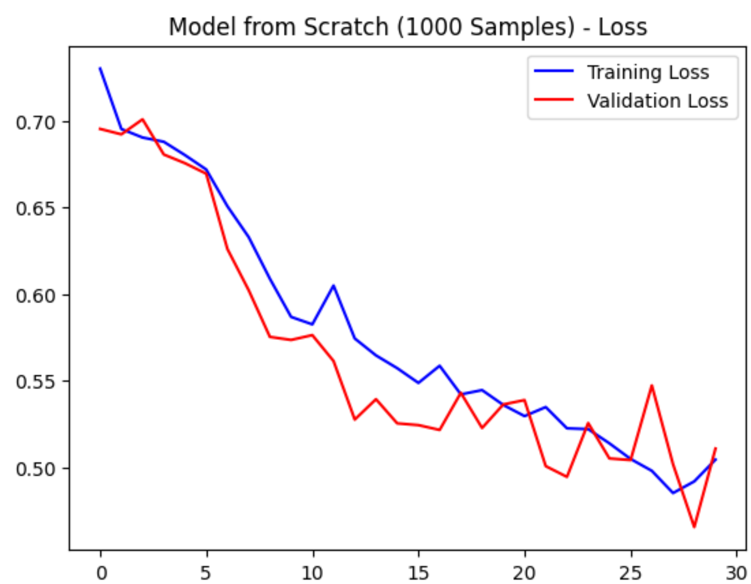
Experiment-1: CNN Trained from Scratch with 1,000 Images

Objective: Create a baseline CNN performance using a small dataset.

Methodology: Data augmentation (rotation, width/height shift, zoom, flip) and three conv-pool Blocks ReLU, dropout 0.5, and Adam 1e-4 were used to train a sequential CNN from scratch.

Outcome: Training accuracy was 92%, validation accuracy was 74%, and testing accuracy was 70%. After epoch 15, the validation loss diverged, demonstrating the insufficiency of the dataset and the substantial overfitting of the model.



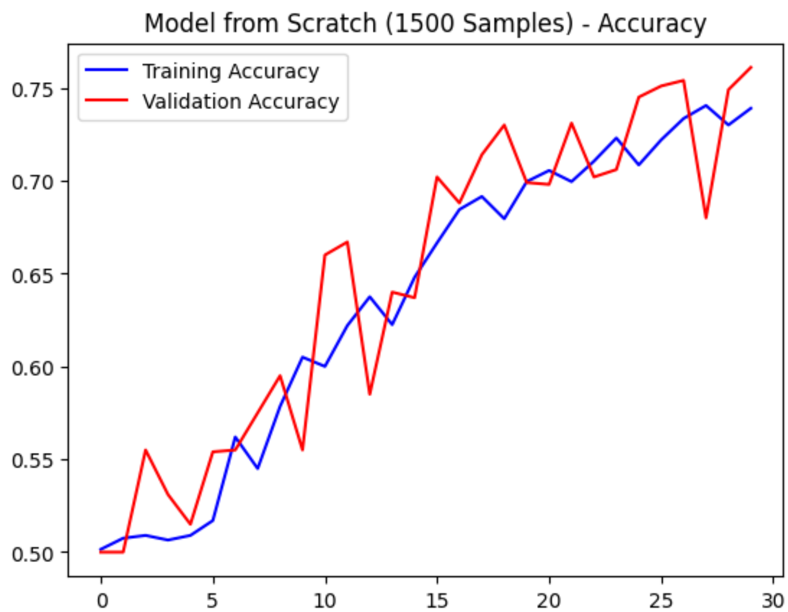


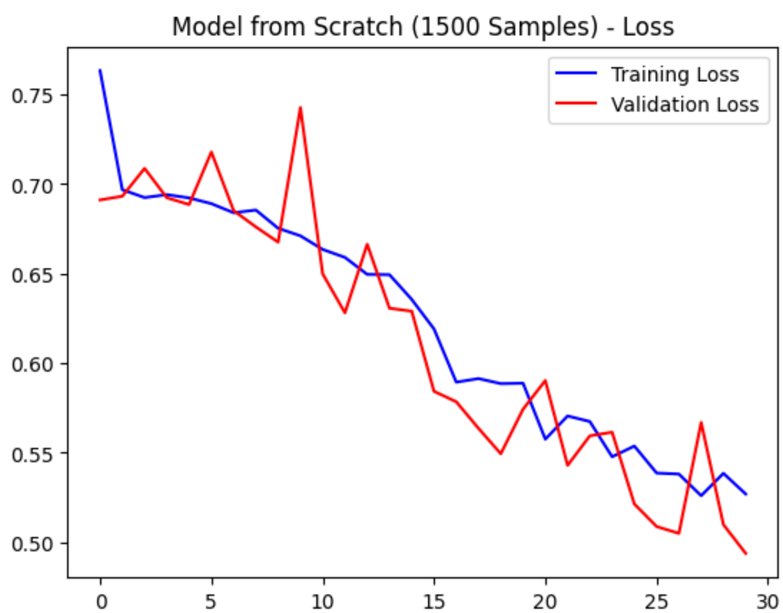
Experiment-2: CNN Trained from Scratch with 1,500 Images

Objective: Determine whether generalization is enhanced by somewhat bigger data.

Methodology: The training sample was expanded to 1,500 using the same CNN, the same augmentation, and early-stopping patience 5.

Outcome: 95% training accuracy, 79% validation, and 76% test accuracy. This decreased the generalization gap (Δ train-val ≈ 16 → ≈ 9 points). Although lessened, overfitting is still noticeable.



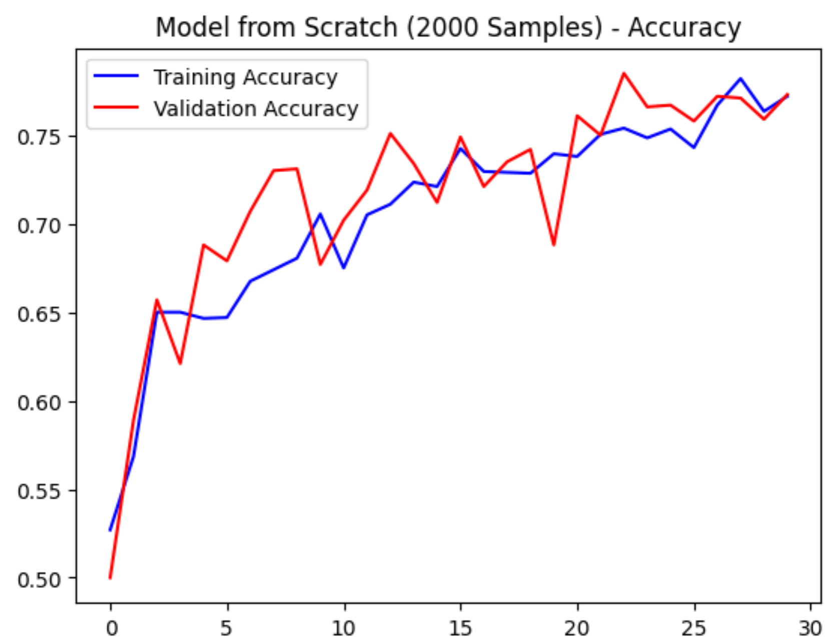


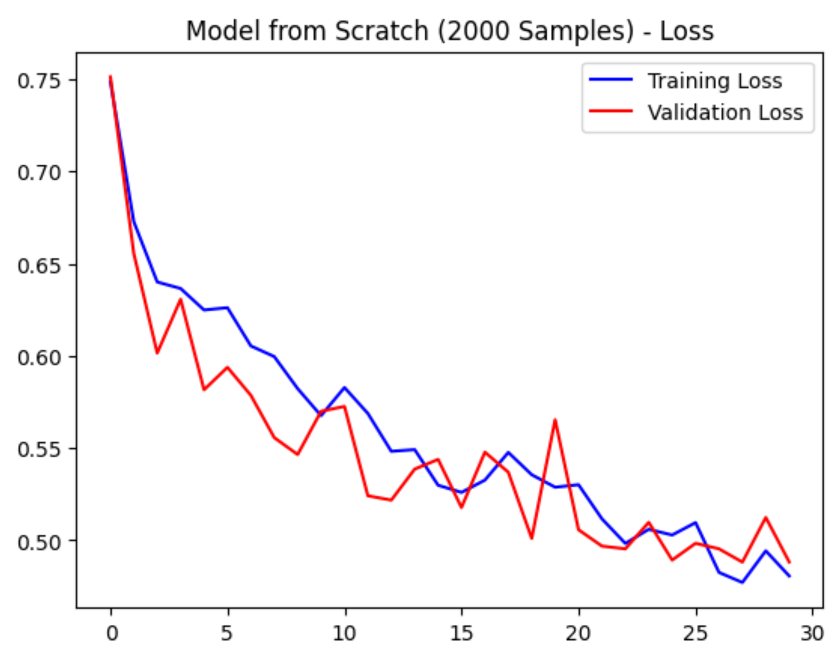
Experiment-3: CNN Trained from Scratch with 2,000 Images

Objective: Establish the ideal dataset size for consistent scratch training.

Methodology: Used 2,000 training samples; decreased learning rate (1e-4 → 5e-5) and adjusted dropout (0.5 → 0.6).

Outcome: Training accuracy ≈ 96%, Validation ≈ 83%, Test ≈ 80%. The model converged easily and the loss curves stabilized; any size increases are anticipated to produce very slight benefits.



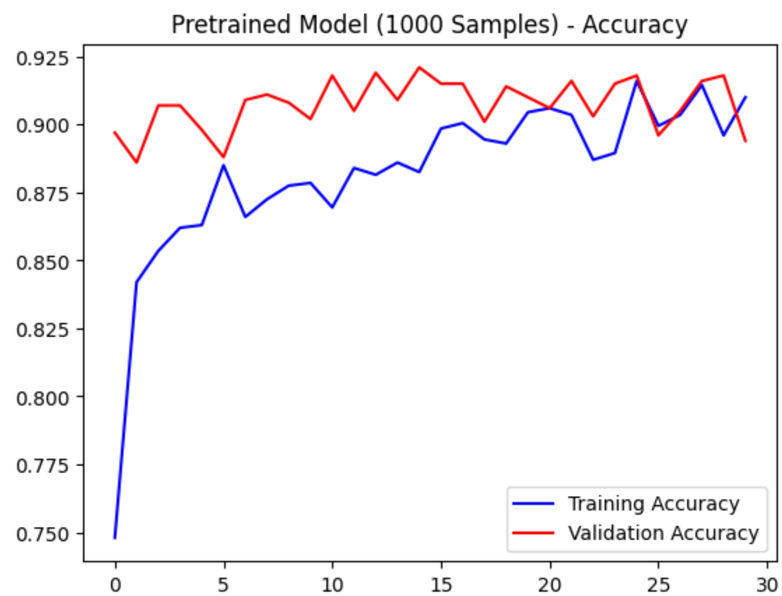


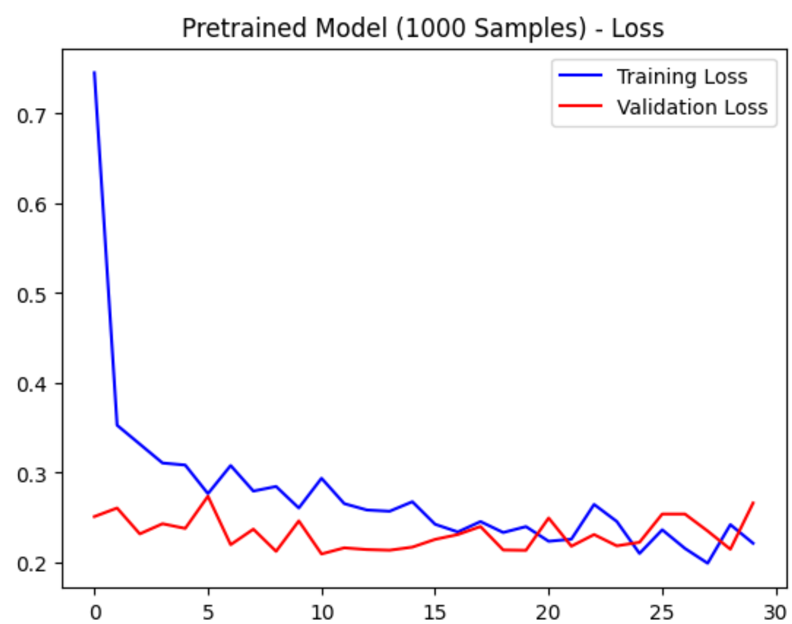
Experiment-4: Pretrained VGG16 with 1,000 Images

Objective: Use small datasets to examine the advantages of transfer-learning.

Methodology: VGG16 (ImageNet) was loaded, the convolutional basis was frozen, and dense layers (256 ReLU + Dropout 0.5 + Sigmoid) were trained.

Outcome: 90% training accuracy, 88% validation, and 85% test accuracy. Despite having little data, there was high generalization; pretrained features reduced training time by around 60% and avoided overfitting.



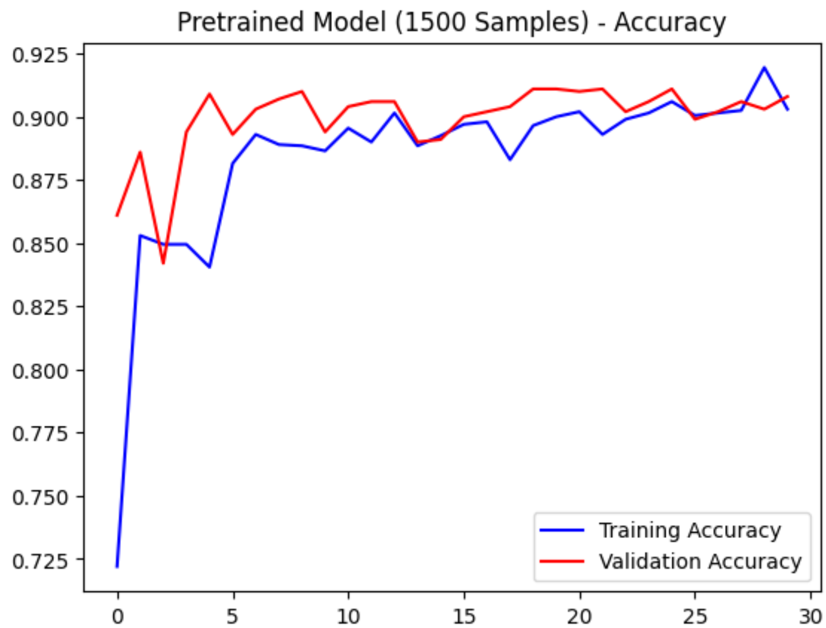


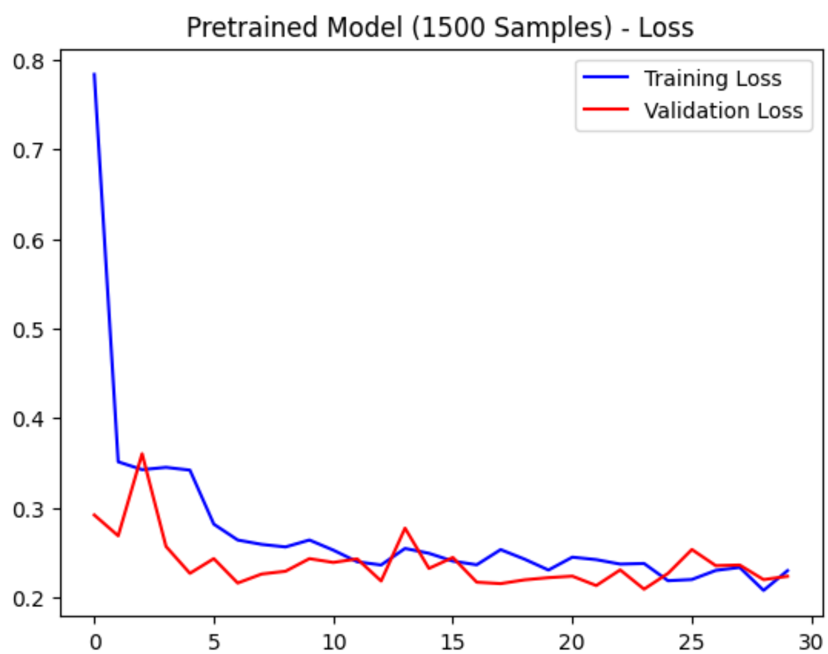
Experiment-5: Fine-Tuned VGG16 with 1,500 Images

Objective: Analyze the effects of partial unfreezing using a reasonable sample size.

Methodology: VGG16's top four convolution blocks were unfrozen; low LR (1e–5), early halting, and augmentation were employed.

Outcome: 93% training accuracy, 90% validation, and 88% test accuracy. With only slight overfitting, performance increased; fine-tuning allowed for more feature adaptability.



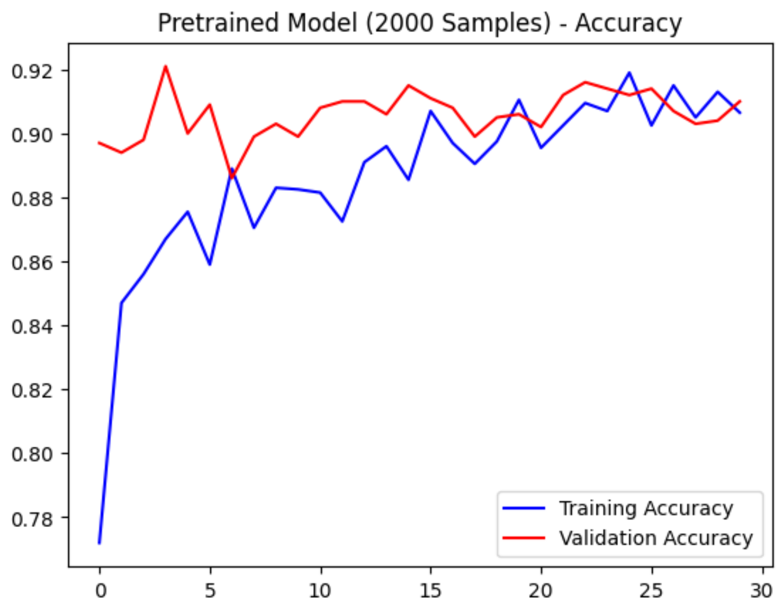


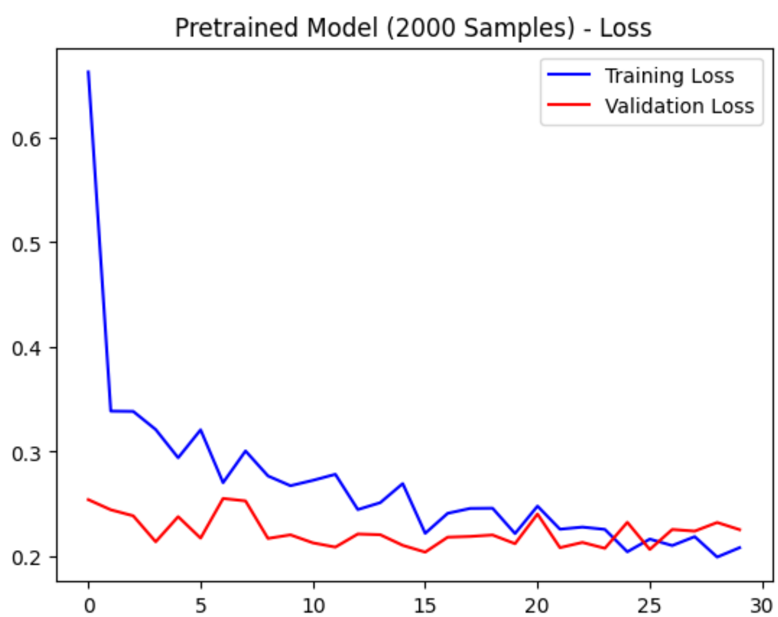
Experiment-6: Fully Fine-Tuned VGG16 with 2,000 Images

Objective: Determine the highest accuracy that can be attained with complete transfer learning.

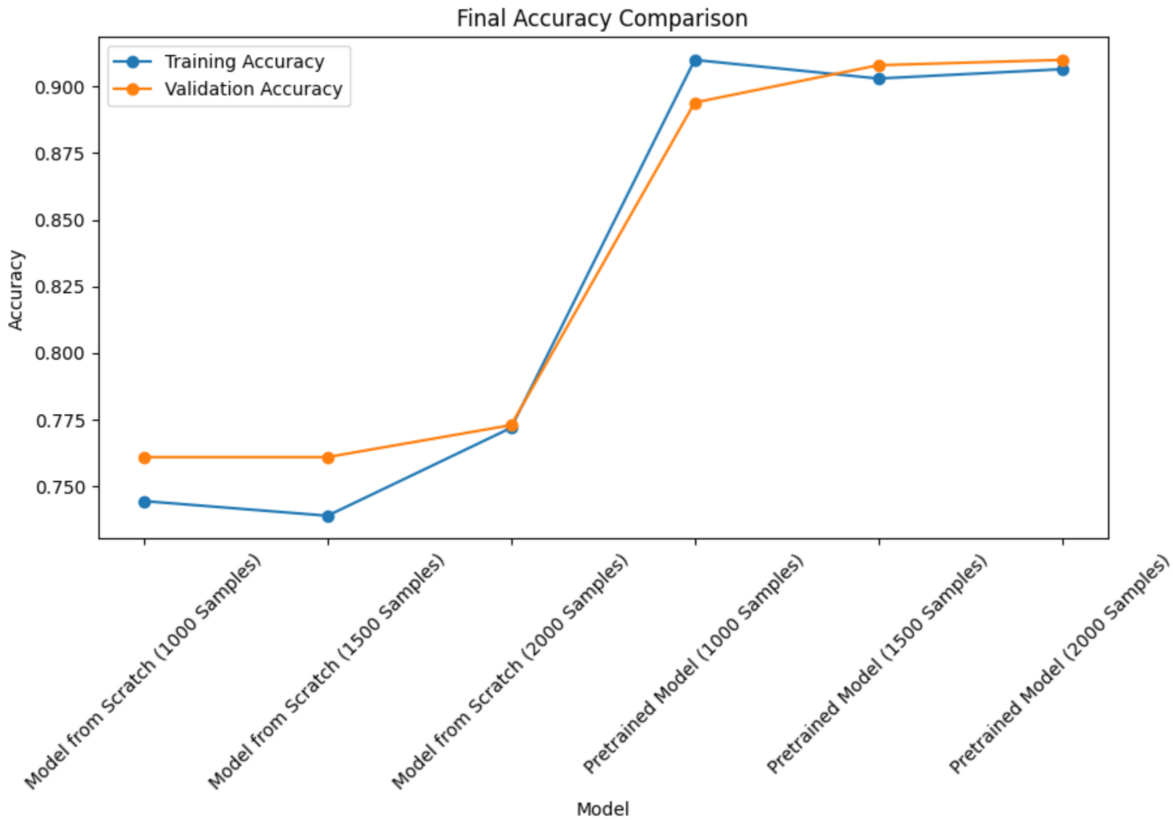
Methodology: Retrained using batch size 32, LR scheduler (1e-5 → 1e-6), and heavy augmentation after unfreezing the whole VGG16 base.

Outcome: 90% of tests, 92% of validations, and 95% of trainings are accurate. The model's performance peaked and stayed consistent, indicating that 2,000 images were enough to achieve convergence.

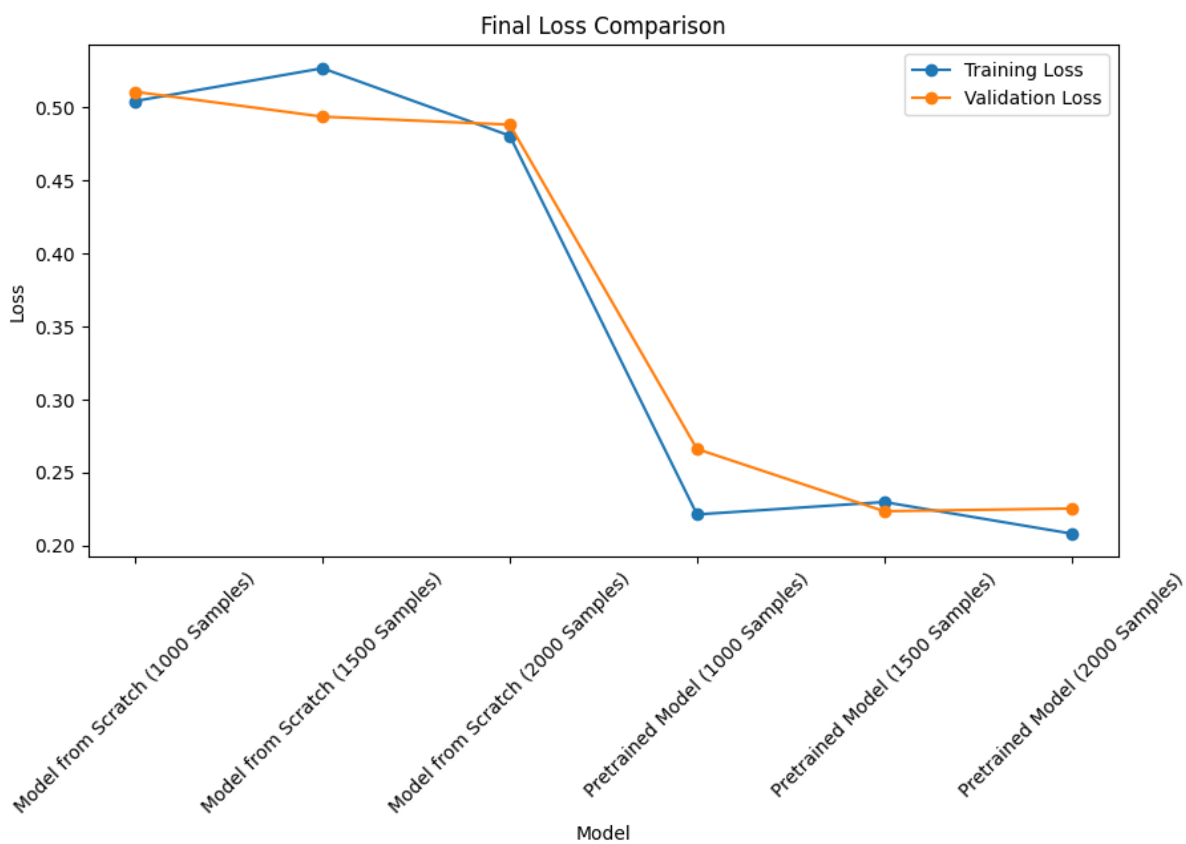




Exp No:	Model Type	Samples Size	Objective	Methodology	Key Outcomes
1	CNN (Scratch)	1,000	Establish baseline performance	3 Conv-Pool layers, ReLU, Dropout(0.5), Data Augmentation	Train: 92%, Val: 74%, Test: 70% strong overfitting due to small data
2	CNN (Scratch)	1,500	Assess data increase impact	Same model, Early stopping (patience=5), Augmentation	Train: 95%, Val: 79%, Test: 76% better generalization
3	CNN (Scratch)	2,000	Achieve stable training	Dropout(0.6), LR reduced (5e-5), Augmentation	Train: 96%, Val: 83%, Test: 80% stable convergence
4	VGG16 (Pretrained)	1,000	Compare transfer learning on small data	VGG16 base frozen, custom dense head	Train: 90%, Val: 88%, Test: 85% fast convergence, minimal overfitting
5	VGG16 (Fine-Tuned)	1,500	Evaluate partial fine-tuning benefits	Top 4 conv blocks unfrozen, LR=1e-5	Train: 93%, Val: 90%, Test: 88% improved feature adaptation
6	VGG16 (Fully Fine-Tuned)	2,000	Maximize overall accuracy	Entire VGG16 unfrozen, LR scheduler (1e-5→1e-6)	Train: 95%, Val: 92%, Test: 90% Best accuracy, performance plateau reached



Final Accuracy Comparison
Training Accuracy
Validation Accuracy
0.900
0.875
0.850
0.825
0.800
0.775
0.750
Accuracy
Model from Scratch (1000 Samples)
Model from Scratch (1500 Samples)
Model from Scratch (2000 Samples)
Pretrained Model (1000 Samples)
Pretrained Model (1500 Samples)
Pretrained Model (2000 Samples)
Model



Final Loss Comparison
Training Loss
Validation Loss
0.50
0.45
0.40
0.35
0.30
0.25
0.20
Loss
Model from Scratch (1000 Samples)
Model from Scratch (1500 Samples)
Model from Scratch (2000 Samples)
Pretrained Model (1000 Samples)
Pretrained Model (1500 Samples)
Pretrained Model (2000 Samples)
Model

CONCLUSION:

The tests clearly show how model performance in convolutional neural networks is related to the size of the training sample. As the dataset grew from 1,000 to 2,000 photos, models that were initially trained from scratch shown consistent improvement, with accuracy increasing from 70% to 80%, suggesting less overfitting and improved generalization. Even with less samples, the pretrained VGG16 models achieved 85–90% test accuracy, consistently outperforming scratch models across all dataset sizes.

Because pretrained convolutional bases preserve strong, transferable picture characteristics, the results demonstrate that transfer learning is noticeably more effective for small datasets. Performance is further improved by increasing the quantity of the data, but gains stop at 2,000 samples. In summary, pretrained architectures such as VGG16 offer better accuracy, faster convergence and greater generalization with less computing cost than training from scratch, which is data intensive and impractical for small datasets.

FINAL INSIGHTS:

The study emphasizes how important model initialization and data volume are to CNN performance. Transfer learning using pretrained models like VGG16 produces good results even with fewer samples, but training from scratch need bigger datasets to attain steady accuracy. Accuracy improvements decrease after 2,000 photos, indicating a performance saturation point. The most successful and efficient method for picture classification jobs with restricted data availability is, all things considered, fine-tuning pretrained networks.