Report

Introduction:

This assignment aims to develop a convolutional neural network (CNN) for classifying cat and dog images by applying convolutional techniques to image data. The objective is to assess how the size of the training dataset affects model performance, and to compare the effectiveness of training a model from scratch versus using a pretrained model.

Convolution: Convolution is a method used in machine learning, especially in image processing, to help a computer identify patterns or features in data, like edges, shapes, or textures in pictures.

It works by sliding a small grid (called a **filter** or **kernel**) over the input image and performing a simple mathematical operation to create a new image that highlights certain features. This process helps the machine learn what's important in an image — for example, the outline of a face or the shape of an object.

By using convolution, computers can focus on important visual features without looking at the entire image at once. This makes it easier and faster for the machine to recognize objects in images, such as detecting cats, cars, or people.

A **Convolutional Neural Network** (CNN) is a type of machine learning model that helps computers understand images by learning to detect patterns and features automatically.

They help computers **see**, **analyze**, and **understand** images by automatically detecting patterns like edges, shapes, or objects — just like the human brain processes visual information.

Problem Definition:

The objective is to perform binary classification by correctly identifying each image as either a 'cat' or a 'dog.' This involves designing and refining CNN architectures to improve accuracy and minimize overfitting.

Data Preparation:

The dataset used for this experiment is the Cats and Dogs dataset.

The data was divided into three subsets: - Training set: **1000 images**

Validation set: **500 images**

Test set: **500 images**

A CNN model was built from the ground up using several Conv2D and MaxPooling2D layers to extract features. To mitigate overfitting, data augmentation methods such as rotation, flipping, and zooming were applied. All images were resized to 150x150 pixels, converted to RGB format, and normalized by scaling pixel values to the [0,1] range to enhance model performance.

Data augmentation is like giving your model "extra practice" by showing it more versions of the same thing, helping it become better at making accurate predictions.

Data augmentation means **creating new versions of your existing images by slightly changing them** — like flipping, rotating, zooming in, or changing brightness — to help a computer model learn better.

Example:

You're teaching a computer to recognize cats and dogs. But you only have 100 pictures of each. Instead of taking more photos, you **make small edits to the ones you already have**

- Turning a cat photo upside down (rotation)
- Flipping a dog photo left to right (horizontal flip)
- Zooming in on a cat's face
- Making a photo brighter or darker

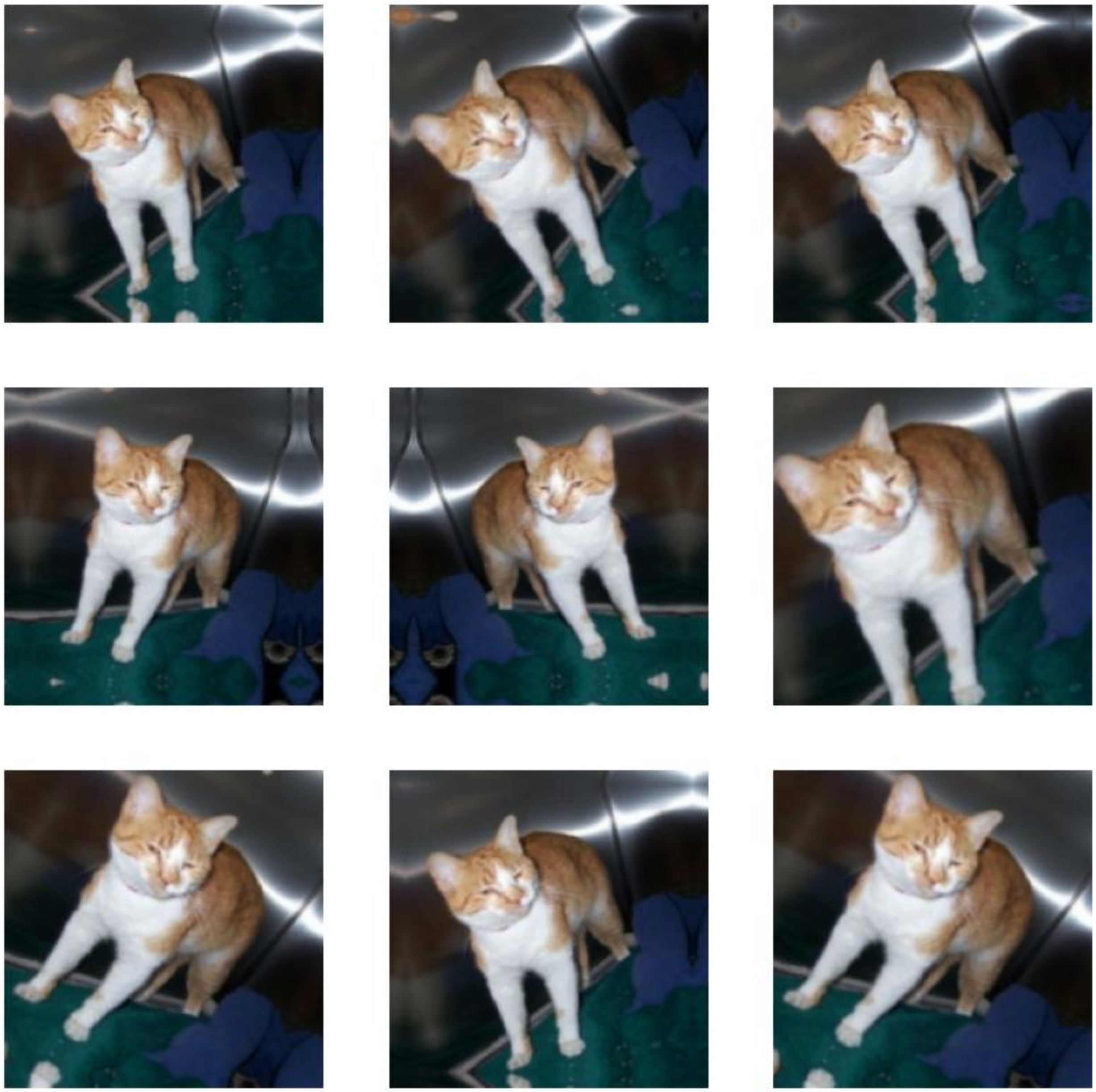
Now, the model sees *more variety*, even though you're still using the same original pictures. It learns to recognize cats and dogs **no matter how they appear** — sideways, close-up, or in different lighting.

Why it helps:

- Makes the model **smarter and more flexible**
- Helps it perform better on **real-world images** it hasn't seen before
- **Reduces overfitting**, which is when the model memorizes the training data instead of learning from it

Steps:

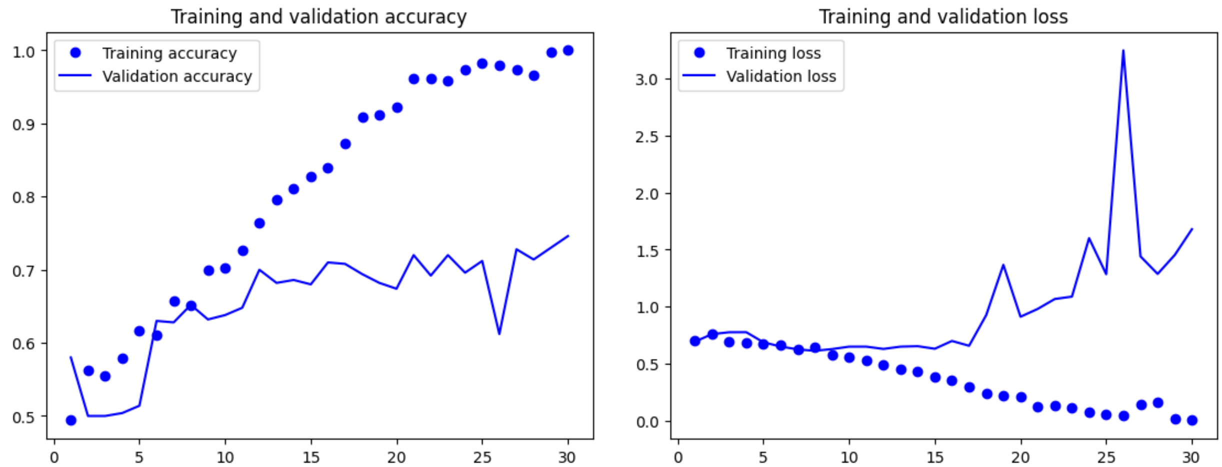
- A data augmentation pipeline is created using Keras, which includes random horizontal flipping, slight rotation, and zooming.
- A batch of images is taken from the training dataset.
- The first image in that batch is augmented 9 times using the defined transformations.
- Each augmented image is displayed in a 3x3 grid using matplotlib for visualization.



Training Samples: 1000(Model-1)

The **Baseline Model-1** delivers a relatively modest performance with a test accuracy of 63.6% and a validation accuracy of 65.2%. The test loss stands at 0.63, while the validation loss is 0.61, reflecting challenges in adapting to unseen data.

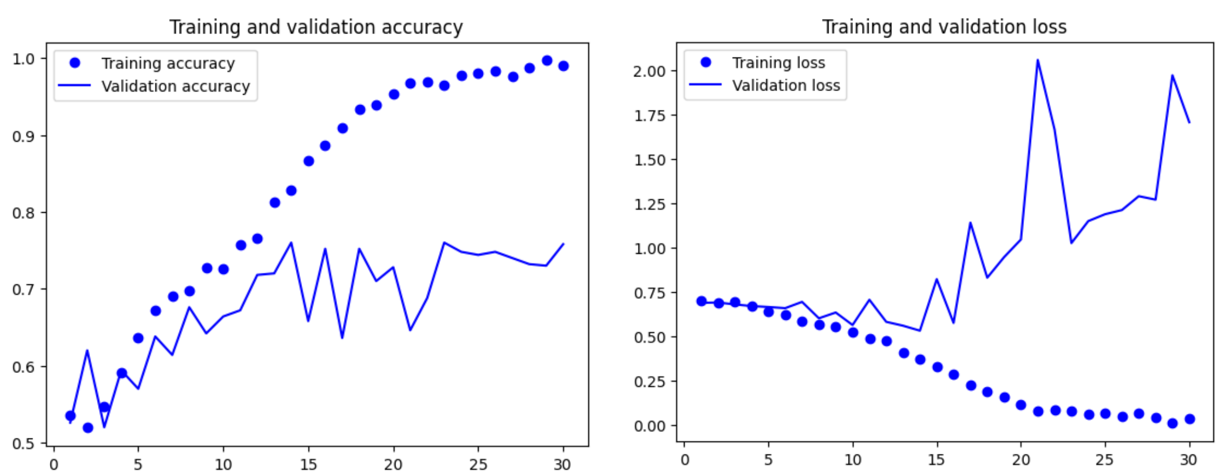
On the other hand, the **Augmentation Model-1** performs significantly better, achieving a test accuracy of 74.0% and a validation accuracy of 75.2%. The validation loss improves to 0.49, showing a considerable drop from the baseline's 0.61. This highlights the effectiveness of data augmentation in reducing overfitting and boosting generalization. Below are the curves of loss and accuracy.



Training Samples: 1500(Model-2)

The **Baseline Model-2** shows improvement compared to models trained on fewer samples, with a test accuracy of 72.6% and a validation accuracy of 76.0%. The test loss of 0.57 and validation loss of 0.53 still indicate moderate overfitting.

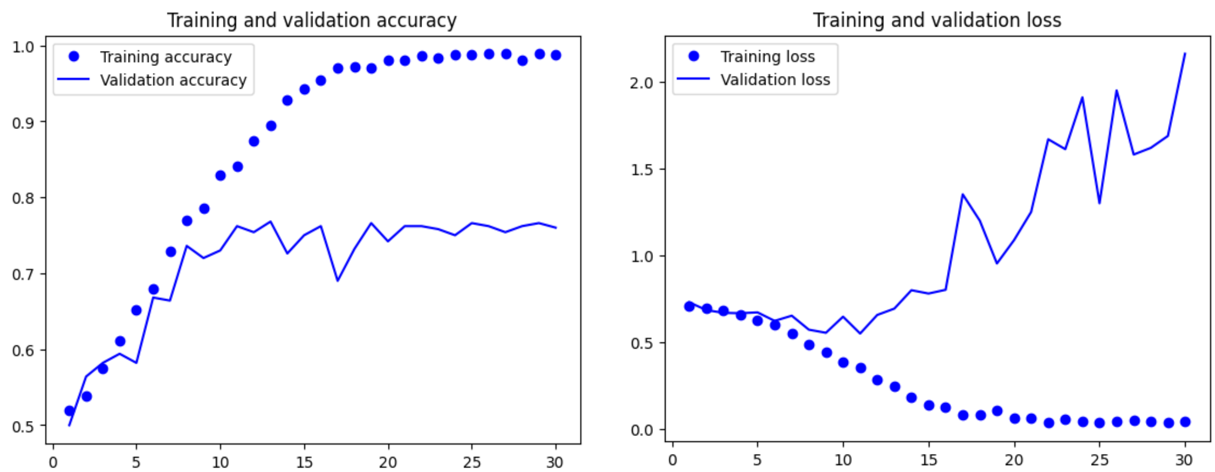
However, the **Augmentation Model-2** demonstrates a strong performance leap, with test accuracy reaching 80.4% and validation accuracy climbing to 83.4%. The validation loss drops from 0.53 to 0.43, indicating a much better generalization ability. These results further confirm the benefit of applying data augmentation techniques, even as training data increases. Below are the curves of loss and accuracy.



Training Samples: 2000(Model-3)

Baseline Model-3 achieves a test accuracy of 73.6% and a validation accuracy of 76.2%, showing decent performance. However, the test loss of 0.66 suggests inefficiencies in capturing testing data patterns, even though validation loss is better at 0.54.

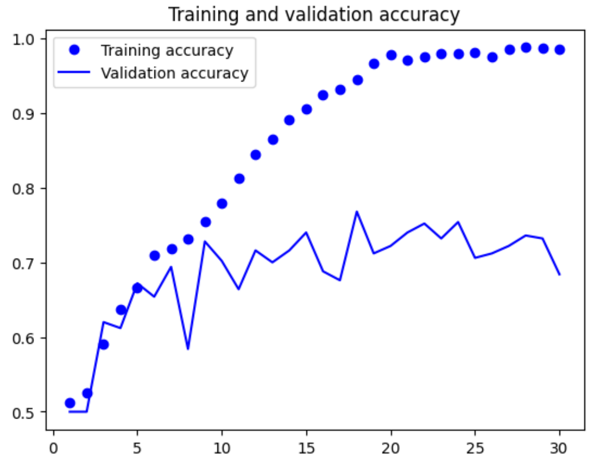
The **Augmentation Model-3** outperforms the baseline with a test accuracy of 83.4% and a validation accuracy of 81.8%. The validation loss remains at 0.45, a significant improvement from the baseline model. This demonstrates that even at higher sample sizes, augmentation continues to help reduce overfitting and enhance overall model robustness. Below are the curves of loss and accuracy.

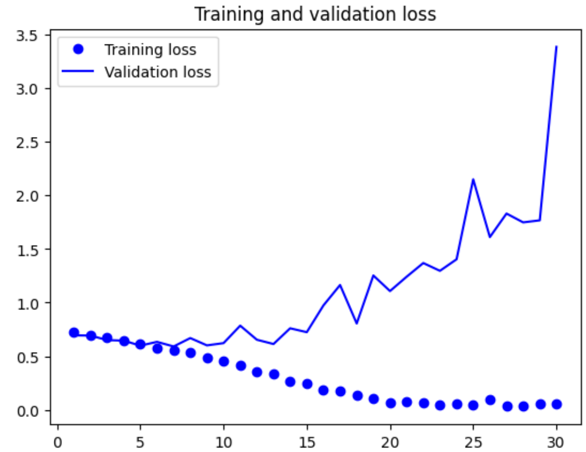


Training Samples: 1700(Model-4)

Baseline Model-4 yields a test accuracy of 67.4% and a validation accuracy of 69.4%, which shows room for improvement. The model ends training with a test loss of 0.66 and a validation loss of 0.59, suggesting some difficulty in adapting to new data.

Augmentation Model-4, by comparison, reaches a test accuracy of 76.2% and a validation accuracy of 78.2%. The validation loss is reduced to 0.45 from the baseline's 0.59, indicating more robust learning and improved generalization. This again supports the argument for using data augmentation, particularly when the dataset is not massive.Below are the curves of loss and accuracy.

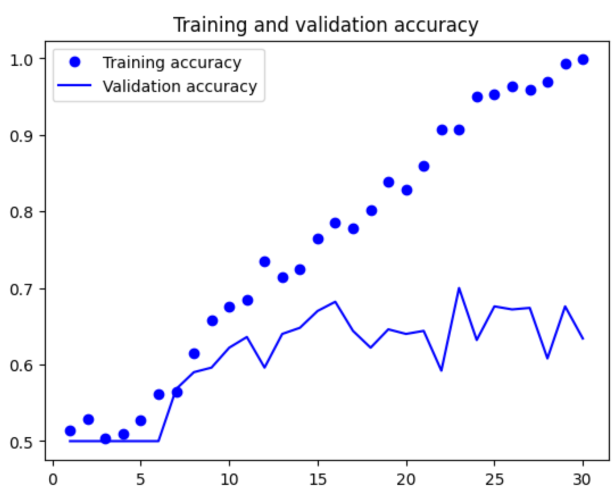


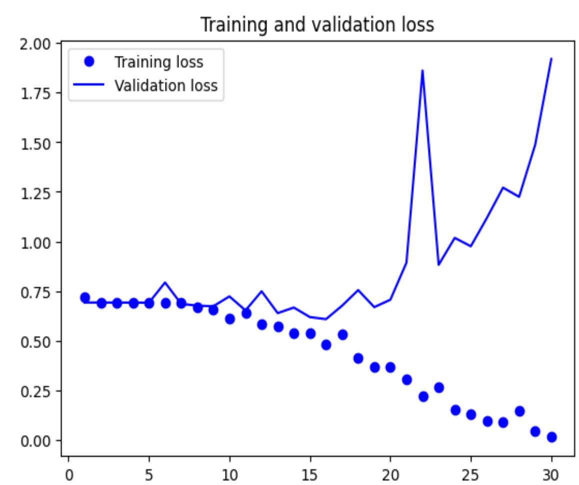


Training Samples: 600(Model-5)

The **Baseline Model-5** performs moderately, with a test accuracy of 65.8% and a validation accuracy of 68.2%. While slightly better than chance, the results suggest the model struggles to capture general patterns in the data. The test loss is 0.67 and the validation loss is 0.60, indicating a potential overfitting issue.

In contrast, the **Augmentation Model-5** shows noticeable improvement, achieving a test accuracy of 71.0% and a validation accuracy of 72.4%. The validation loss decreases from 0.60 to 0.54, suggesting better generalization. This confirms that data augmentation positively impacts model performance under limited data conditions. Below are the curves of loss and accuracy.





Training Sample	Models Type	Test Accuracy	Test Loss	Validation Accuracy	Validation Loss
1000	Baseline Model-1	63.6	0.63	65.2	0.61
1000	Augmentation Model-1	74.0	0.51	75.2	0.49
1500	Baseline Model-2	72.6	0.57	76.0	0.53
1500	Augmentation Model-2	80.4	0.47	83.4	0.43
2000	Baseline Model-3	73.6	0.66	76.2	0.54
2000	Augmentation Model-3	83.4	0.45	81.8	0.45
1700	Baseline Model-4	67.4	0.66	69.4	0.59
1700	Augmentation Model-4	76.2	0.51	78.2	0.45
600	Baseline Model-5	65.8	0.67	68.2	0.60
600	Augmentation Model-5	71.0	0.63	72.4	0.54

Key Findings:

1. Impact of Data Augmentation

- Across all training sample sizes, **augmentation significantly improves model performance**.
- On average, **augmented models outperform baseline models** by **~7–10%** in test and validation accuracy.
- Augmentation consistently lowers both **test and validation loss**, suggesting better generalization.

2. Impact of Training Sample Size

- **Larger training sizes generally yield better results**, but with diminishing returns.
  - From 600 to 1500 samples, there is a **notable gain** in accuracy.
  - Beyond 1500–1700 samples, **gains are smaller** unless paired with augmentation.
- Baseline models see **less consistent improvements** with increased sample size compared to augmented models.

   - E.g., Baseline Model-1 (1000 samples) has lower accuracy than Baseline Model-5 (600 samples), showing instability without augmentation.

3. Best Performing Model

- **Augmentation Model-3 (2000 samples)** shows the **highest test accuracy (83.4%)** and the **lowest test loss (0.45)**, confirming that **data quantity + augmentation = optimal performance**.

Conclusion:

- **Data Augmentation is more influential than sample size alone** in improving model performance.
- **Combining larger datasets with augmentation provides the best results**, as seen in Augmentation Model-3.
- Baseline models, especially with fewer samples, tend to **underfit or generalize poorly**.
- Augmentation helps simulate data diversity and **mitigates overfitting**, especially in small to mid-sized datasets.

Pretrained Network:

Using the VGG16 architecture alongside data augmentation techniques leverages pretrained features from ImageNet. The application of transformations like rotations and flips introduces greater variability in the input data, which helps the model generalize better and reduces the risk of overfitting. As a result, this approach improves classification accuracy and enhances the model's robustness, particularly in tasks such as distinguishing between cats and dogs.

Starting with **Model 6** trained with **1000 samples**, pretraining with data augmentation gives a **test accuracy of 96.8%** and **validation accuracy of 98.0%**. The **test loss** was **7.2** and the **validation loss** was **1.81**. Performance on a small dataset is reasonable; augmentation must have helped in preventing overfitting in this case with such limited data.

Training with **1500 samples** on **Model 7** increased the **test accuracy to 97.4%**, while **validation accuracy** remained slightly lower at **97.6%**. However, the **test loss** improved to **3.66**, and the **validation loss** remained at **1.9**. The accuracy increased with more samples, yet the relatively higher losses compared to accuracy gains may indicate some variance or mild instability, suggesting a trade-off even with augmentation in place.

Model 8 had the **highest validation accuracy** at **98.2%** and matched the highest **test accuracy** at **97.4%** using **2000 training samples**. The **test loss** was the **lowest among all** at **3.45**, and the **validation loss** stayed close at **1.95**. This indicates strong generalization and effective learning.

With the largest sample size, augmentation continues to aid learning, although gains appear to plateau slightly beyond this point.

Model 9 presented the best overall balance when trained with **1700 samples**. It achieved a **test accuracy of 97.0%** and **test loss of 3.92**, which is still among the lowest. The **validation accuracy of 97.8%** and **validation loss of 1.86** reinforce this model's generalization ability. It stands out for balancing accuracy and loss effectively without overfitting, making it a strong candidate for real-world performance.

Concluding with **Model 10**, trained on only **600 samples**, performance notably dropped. The **test accuracy fell to 95.6%**, the **lowest** among all models, and the **test loss rose sharply to 11.26**. Despite this, the **validation accuracy was surprisingly high at 98.2%**, with a **validation loss of 2.63**. This disparity suggests overfitting and poor generalization — a clear indication that the sample size was insufficient, even with augmentation.

Summary:

Pretraining with data augmentation improved model performance across all sample sizes. However, there appears to be a sweet spot in sample size — **Model 9 with 1700 samples** that strikes the best balance between accuracy, loss, and generalization. Too few samples, as seen in Model 10, result in unreliable models for deployment. Conversely, beyond a certain point, increasing the sample size (e.g., Model 8) yields diminishing returns. Thus, **Model 9 stands out as the optimal choice**, offering high performance and robust generalization for deployment scenarios.

Training Sample	Models Type	Optimization technique	Test Accuracy	Test Loss	Validation Accuracy	Validation Loss
1000	Model 6	Data Augmentation	96.8	7.2	98.0	1.81
1500	Model 7	Data Augmentation	97.4	3.66	97.6	1.92
2000	Model 8	Data Augmentation	97.4	3.45	98.2	1.95
1700	Model 9	Data Augmentation	97.0	3.92	97.8	1.86
600	Model 10	Data Augmentation	95.6	11.26	98.2	2.63

1. Effectiveness of Pretrained VGG16 with Augmentation

- All models achieved **very high accuracy** (≥95.6% test accuracy), confirming the strength of using pretrained features from VGG16.

- Data augmentation further improved generalization by introducing variability, reducing overfitting, and helping models adapt better to unseen data.

2. Impact of Training Sample Size

- Performance remains **consistently high across all sample sizes**, indicating that the pretrained model is highly effective even with **smaller datasets** (as low as 600).
- Smaller datasets (like Model 10 with 600 samples) still achieved **98.2% validation accuracy**, showing that **pretraining compensates for limited data** when paired with augmentation.

3. Test vs. Validation Consistency

- Models showed **balanced performance** on both test and validation sets, suggesting **good generalization**.
- Test losses are slightly higher than validation losses in some cases, but accuracy remains stable.

4. Best Performing Model

- **Model 8 (2000 samples)** and **Model 7 (1500 samples)** achieved the **highest test accuracy (97.4%)**.
- **Model 8** had the **lowest test loss (3.45)** and **highest validation accuracy (98.2%)**, indicating overall best performance.

Conclusion:

- **Using a pretrained VGG16 network with data augmentation results in significantly better performance** compared to training from scratch, especially with limited data.
- Pretrained models are **less dependent on large datasets**, performing well even with smaller sample sizes.
- Data augmentation enhances model robustness and supports generalization by simulating real-world variability.
- Overall, this approach is highly effective for binary classification tasks like Cats vs. Dogs.

Training from scratch vs Pretrained Models

Training Sample	Models Type	Optimization technique	Test Accuracy	Test Loss	Validation Accuracy	Validation Loss
2000	Model 3	Data Augmentation	83.4	0.45	81.8	0.45
1500	Model 7	Data Augmentation	97.4	3.66	97.6	1.92
2000	Model 8	Data Augmentation	97.4	3.45	98.2	1.95

- **Pretrained Models Dramatically Outperform** the scratch model in both **test and validation accuracy**:
  - Pretrained Model 7 (with fewer samples than Model 3) reaches **97.6% validation accuracy** vs **81.8%** for the scratch model.
  - Model 8 (same sample size as Model 3) reaches **98.2%**, confirming the strength of pretrained features.
- **Generalization and Efficiency**:
  - Pretrained models clearly **generalize better**, even with fewer training samples.
  - They benefit from learned representations, allowing them to converge faster and more accurately than models trained from scratch.

Conclusion:

This comparison confirms that **pretrained models** even when trained on fewer samples **significantly outperform scratch-trained models** in image classification tasks. The use of transfer learning allows pretrained models to extract complex features more effectively, improving generalization and reducing the need for large labeled datasets.

In real-world applications where time, data, and computational resources are limited, **pretrained networks with data augmentation** present a **far more efficient and scalable solution** than training from scratch.