The main objective of this project was to classify images of dogs and cats using **Transfer Learning**. Pre-trained CNN models (VGG16 and ResNet50) were used with two approaches:

1. **Feature Extraction** – using the pre-trained model as a fixed feature extractor.
2. **Fine-Tuning** – unfreezing some of the top layers of the pre-trained model to adapt it to the new dataset.

This approach reduces training time, leverages learned features from large datasets like ImageNet, and improves model performance.

2. Dataset Preparation

- Dataset: Cats vs Dogs (~3000 images)
    - Training: 2000 images
    - Validation: 1000 images
- **Preprocessing steps:**

1. **Resize images** to 224×224 pixels to match the input requirements of pre-trained models.
2. **Normalize pixel values** to [0,1] for faster convergence.
3. **Data Augmentation:**
    - Horizontal flips
    - Rotation
    - Zoom
    - Purpose: Increase dataset variability and prevent overfitting.

Understanding:

- CNNs learn better when images are normalized and augmented.
- Augmentation simulates more data, which is helpful for small datasets.

3. Model Development

3.1 Feature Extraction

- Pre-trained model is loaded **without top classifier layers**.

- All base layers are **frozen** → weights not updated.
- A **new classifier** is added on top:
  - GlobalAveragePooling2D → Dense(128, ReLU) → Dropout(0.3) → Dense(1, Sigmoid)
- Only these new layers are trained on the dataset.

Purpose:

- Use existing CNN features to extract important patterns.
- Faster training with fewer parameters to update.

3.2 Fine-Tuning

- Unfreeze the last few layers of the base model.
- Train both the **unfrozen layers** and the new classifier.

Purpose:

- Allow the model to adapt high-level features to the new dataset.
- Can improve accuracy if enough data is available.

4. Training & Evaluation

- All models trained on Google Colab GPU.
- **Epochs:** 2 (for demonstration / time-saving)
- **Steps per epoch:** 50
- Training metrics: **accuracy and loss** recorded for each epoch.
- Models saved as .h5 files for later use.

4.1 Results

Model	Approach	Accuracy (Validation)	Loss (Validation)
VGG16	Feature Extraction	0.8547	0.4344
VGG16	Fine-Tuning	0.8266	0.4536
ResNet50	Feature Extraction	0.6078	0.6678
ResNet50	Fine-Tuning	0.5844	0.6650

Observations:

1. VGG16 outperformed ResNet50 on this dataset.
2. Feature Extraction achieved slightly higher validation accuracy than Fine-Tuning for VGG16.
3. ResNet50 performed worse, possibly due to smaller dataset and higher model complexity.

Understanding:

- Smaller datasets favor **Feature Extraction** because fewer parameters are trained.
- Fine-Tuning may require **more data** to improve over Feature Extraction.

5. Conclusion

1. **Transfer Learning is effective** for image classification tasks like dogs vs cats.
2. **Feature Extraction** is fast and works well with small datasets, leveraging learned features from large datasets.
3. **Fine-Tuning** can improve accuracy for larger datasets but may overfit if data is limited.
4. **VGG16** is more suitable than ResNet50 for this small dataset due to better feature extraction and simpler architecture.
5. Preprocessing and data augmentation are essential to improve generalization.

Final takeaway:

- For small datasets, use **Feature Extraction with a pre-trained CNN and a custom classifier**.
- Fine-Tuning is optional but useful if more data is available.
- Transfer Learning drastically reduces training time while achieving good accuracy.

6. Deliverables for Submission

1. **Trained Models**:
    - vgg16_feature.h5

- vgg16_fine.h5
- resnet50_feature.h5
- resnet50_fine.h5

2. **Plots**: Training/Validation accuracy & loss for all 4 models.
3. **Report**: This document summarizing objectives, methods, results, observations, and conclusions.