Transfer Learning with VGG16 on CIFAR-10 Subset

EFFICIENT IMAGE CLASSIFICATION USING PRE-TRAINED MODELS

GROUP PRESENTATION BY

- Sridhar Guggilla
- Ramesh Kelavath
- Hrushitha Nidikonda
- Syam Kolapalli
- Gopi Reddy
- Sneha Ellanki
- Ashritha Bhupathi
- Alekya Saidam

Overview

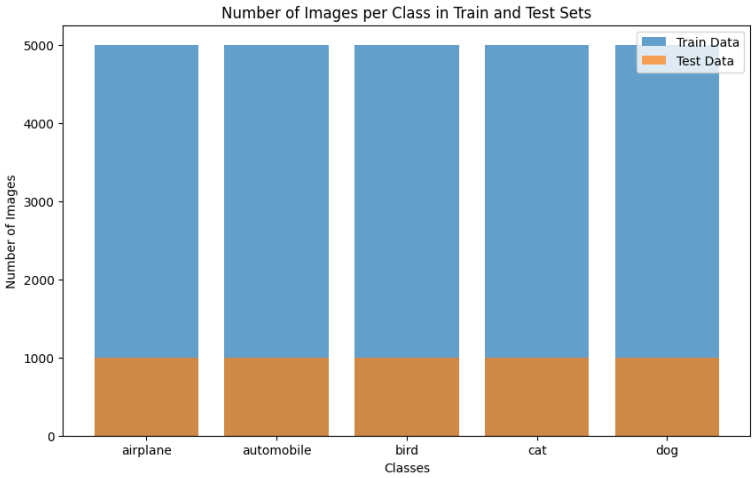
- Goal: Classify a subset of CIFAR-10 images using transfer learning.
- Why Transfer Learning?
- Leverages pre-trained models (e.g., VGG16) for efficiency.
- Reduces computation and training time.
- Improves performance with limited data.

Dataset Overview

- CIFAR-10 Dataset:
- Contains 30,000 32x32 color images across 10 classes.
- 50,000 training images, 10,000 testing images.
- Subset Selected:
- 5 Classes: Airplane, Automobile, Bird, Cat, Dog.
- Why These Classes?
- Diverse objects to test generalization.
- Balances animate and inanimate categories.

Data Preprocessing

- Steps:
- 1. Filtered dataset to focus on 5 relevant classes.
- 2. Normalized images to [0, 1] for stable and efficient training.
- 3. Label Remapping: Adjusted labels to align with model outputs.
- 4. Data Augmentation: Increased data diversity using transformations.
- Challenges:
- Ensuring balanced class distribution.
- Avoiding overfitting despite a smaller dataset.



Model Architecture

Base Model: VGG16 pre-trained on ImageNet.

Custom Layers:

- Global Average Pooling: Reduces dimensionality while retaining features.

- Fully Connected Layer (256 neurons): Adds task-specific learning.

- Dropout (50%): Prevents overfitting.

- Softmax Layer: Outputs probabilities for 5 classes.

Compilation:

- Optimizer: Adam for adaptive learning rates.

- Loss Function: Categorical crossentropy.

- Metrics: Accuracy to track model performance.

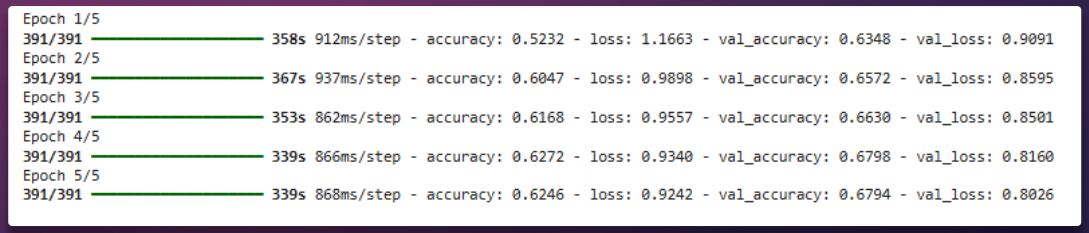
Model: "sequential_3"

Layer (type)	Output Shape	Param #
vgg16 (Functional)	(None, 1, 1, 512)	14,714,688
global_average_pooling2d_3 (GlobalAveragePooling2D)	(None, 512)	0
dense_6 (Dense)	(None, 256)	131,328
dropout_3 (Dropout)	(None, 256)	0
dense_7 (Dense)	(None, 5)	1,285

Total params: 14,847,301 (56.64 MB)
Trainable params: 132,613 (518.02 KB)
Non-trainable params: 14,714,688 (56.13 MB)

Training

- Training Details:
- Batch Size: 64, Epochs: 5.
- Data Augmentation: Generated variations to improve generalization.
- Validation Set:
- Monitored overfitting during training.
- Ensured the model performs well on unseen data.
- Challenges:
- Fine-tuning hyperparameters to balance training and validation performance.

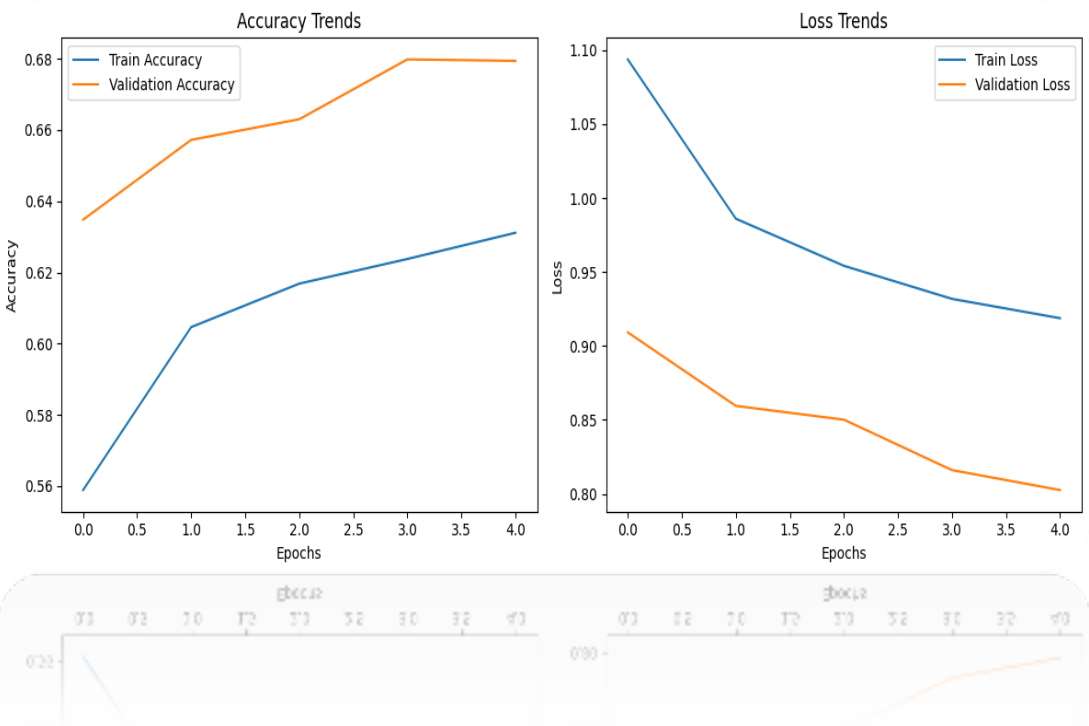


Performance Metrics

- Accuracy and Loss Trends:
- Help diagnose overfitting and underfitting.

Key Insights:

- Consistent validation performance indicates good generalization.
- Diverging trends signal overfitting or model instability.



Model Evaluation

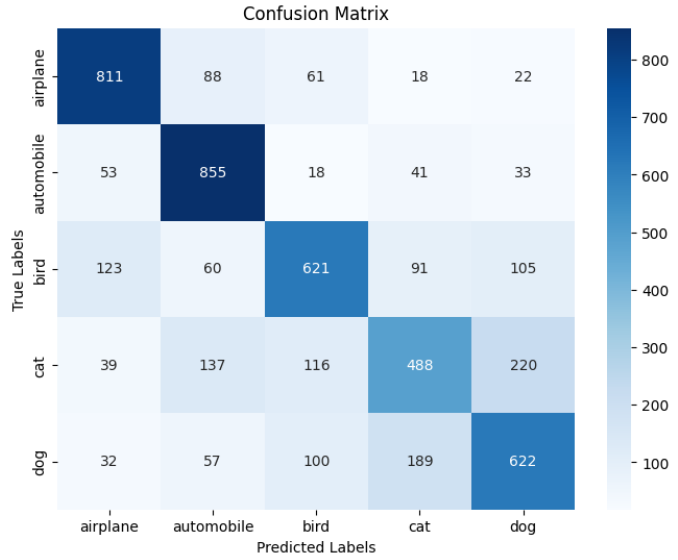
- 1. Classification Report:
- Precision, Recall, F1-score for each class.
- 2. Confusion Matrix:
- Diagonal values: Correct predictions.
- Off-diagonal values: Misclassifications.
- Insights:
- Identifies challenging classes.
- Guides future optimization.

```
157/157 ━━━━━━━━━━━━━━━━━━━━ 56s 357ms/step

Classification Report:
              precision    recall  f1-score   support

    airplane       0.77      0.81      0.79      1000
  automobile       0.71      0.85      0.78      1000
        bird       0.68      0.62      0.65      1000
         cat       0.59      0.49      0.53      1000
         dog       0.62      0.62      0.62      1000

    accuracy                           0.68      5000
   macro avg       0.67      0.68      0.67      5000
weighted avg       0.67      0.68      0.67      5000
```



Model Deployment

- Saved Model:
- Saved as 'cifar10_reduced_vgg16.h5'.
- Benefits:
- Enables inference, fine-tuning, and reproducibility.
- Prediction Function:
- Accepts new images for class prediction.
- Applications:
- Automated image classification systems.
- Benchmarks for other transfer learning tasks.

Conclusion

- Key Achievements:
- Successfully classified a subset of CIFAR-10 using VGG16.
- Demonstrated the efficiency of transfer learning for image recognition.
- Challenges:
- Balancing dataset diversity and model complexity.
- Preventing overfitting with limited data.
- Future Directions:
- Experimenting with other pre-trained models (e.g., ResNet, EfficientNet).
- Hyperparameter tuning for enhanced performance.