

# Transfer Learning for Object Recognition on the Caltech-101 Dataset: A Comparative Analysis of MobileNet and ResNet

Tanish S.K. (CSE-C), Thiagarajan P.K. (CSE-C)  
2nd Year, Department of Computer Science and Engineering  
SSN College of Engineering  
Chennai, India

GitHub: [https://github.com/Thiyaga1586/Transfer\\_Learning\\_for\\_Object\\_Recognition](https://github.com/Thiyaga1586/Transfer_Learning_for_Object_Recognition)

**Abstract**—Object recognition remains a fundamental challenge in computer vision, often requiring massive datasets and extensive computational resources to train deep learning models from scratch. Transfer learning addresses this by leveraging models pre-trained on large-scale datasets. This paper presents an end-to-end pipeline for object classification on the Caltech-101 dataset utilizing transfer learning. A comparative analysis of two popular architectures is conducted: ResNet50, known for its deep residual learning framework, and MobileNetV2, optimized for mobile and resource-constrained environments. Furthermore, two distinct transfer learning strategies are evaluated: freezing the base layers for feature extraction versus fine-tuning the entire network. The methodology encompasses dataset preparation, pretrained model loading, feature extraction, fine-tuning, and rigorous testing. The findings provide insights into the trade-offs between model complexity, computational efficiency, and classification accuracy when adapting pre-trained models to new, domain-specific tasks.

**Index Terms**—Transfer Learning, Object Recognition, Caltech-101, Deep Learning, ResNet50, MobileNetV2, Fine-tuning, Feature Extraction

## I. INTRODUCTION

Deep Convolutional Neural Networks (CNNs) have revolutionized computer vision, achieving state-of-the-art performance in object recognition tasks. However, training these deep networks from scratch requires vast amounts of labeled data and significant computational power, which are not always available. Transfer learning mitigates this by utilizing models pre-trained on massive datasets (like ImageNet) and adapting them to target datasets with limited samples.

The Caltech-101 dataset, comprising images of objects belonging to 101 categories, serves as an excellent benchmark for evaluating the efficacy of transfer learning. While the dataset is diverse, the number of images per category is relatively small, making it an ideal candidate for pre-trained model adaptation rather than from-scratch training.

An end-to-end pipeline is designed and implemented herein to classify objects within the Caltech-101 dataset. Two renowned architectures are specifically compared:

- **ResNet50:** Utilizes residual connections to train highly deep networks without suffering from the vanishing gradient problem.
- **MobileNetV2:** Employs depthwise separable convolutions and inverted residuals to build lightweight deep neural networks.

The impact of two transfer learning paradigms is investigated: freezing the pre-trained weights to utilize the model as a fixed feature extractor, versus fine-tuning the network by updating the weights of the deeper layers alongside the newly added classification head.

## II. LITERATURE SURVEY

The advent of deep learning has drastically improved the performance of image classification systems. Early breakthroughs like AlexNet and VGG established the standard for hierarchical feature extraction. However, as networks grew deeper, optimizing them became challenging. He et al. introduced Residual Networks (ResNet), incorporating skip connections that allowed for the successful training of exceedingly deep models, leading to significant accuracy improvements on benchmarks like ImageNet.

Simultaneously, there has been a growing need to deploy these powerful models on edge devices. Howard et al. proposed MobileNet, which replaces standard convolutions with depthwise separable convolutions, drastically reducing the number of parameters and computational cost while maintaining competitive accuracy.

Transfer learning has been widely adopted to leverage these architectures for specialized tasks. Studies indicate that the early layers of CNNs learn generic features (like edges and textures), which are transferable across domains, while deeper layers learn task-specific features. Research demonstrates that fine-tuning deeper layers often yields better performance than mere feature extraction, though it requires more careful hyperparameter tuning to avoid overfitting, especially on smaller datasets like Caltech-101.

### III. METHODOLOGY

The methodology follows a systematic end-to-end pipeline: Dataset Preparation → Pretrained Model Loading → Feature Extraction → Fine-Tuning → Testing.

#### A. Dataset Preparation

The Caltech-101 dataset consists of images across 101 object categories, plus a background category. Pre-processing involves:

- **Resizing:** Standardizing all input images to dimensions compatible with the pre-trained models (e.g.,  $224 \times 224$  pixels).
- **Normalization:** Scaling pixel values using the mean and standard deviation of the ImageNet dataset to align with the pre-trained weights.
- **Data Splitting:** Dividing the dataset into training, validation, and testing sets using stratified sampling to maintain class balance.
- **Data Augmentation:** Applying random rotations, flips, and zooms to the training set to improve model generalization.

#### B. Pretrained Model Loading

Base models (ResNet50 and MobileNetV2) pre-trained on ImageNet are imported. The top classification layers (the fully connected layers) are discarded, as they are specific to ImageNet's 1000 classes. A new global average pooling layer followed by a dense layer with a softmax activation corresponding to the 102 classes of Caltech-101 is appended.

#### C. Feature Extraction (Freezing Layers)

In the first experimental setup, the weights of the entire base model are frozen. During training, only the weights of the newly added dense layer are updated. This approach treats the pre-trained network as a static feature extractor. A standard learning rate and the Adam optimizer are utilized.

#### D. Fine-Tuning

In the second setup, the top layers of the base model are unfrozen. This allows the network to adapt higher-order feature representations to the specific characteristics of the Caltech-101 dataset. Fine-tuning is performed utilizing a significantly lower learning rate to prevent catastrophic forgetting of the pre-trained knowledge.

### IV. RESULTS AND DISCUSSION

The end-to-end pipelines for both ResNet50 and MobileNetV2 were evaluated on the Caltech-101 validation set. The primary metrics utilized for comparison were classification accuracy and validation loss.

#### A. Performance Comparison: ResNet50 vs. MobileNetV2

The quantitative results of the fine-tuned models are summarized in Table I.

TABLE I  
VALIDATION PERFORMANCE ON CALTECH-101

Model Architecture	Validation Accuracy	Validation Loss
ResNet50	94.14%	0.217
MobileNetV2	93.87%	0.212

#### B. Analysis of Findings

- **Accuracy Advantage of ResNet50:** ResNet50 achieved a slightly higher validation accuracy (94.14%) compared to MobileNetV2 (93.87%). This aligns with theoretical expectations, as ResNet50's deeper architecture and residual connections allow for the capture of highly complex, hierarchical feature representations, yielding a marginal edge in raw predictive power.
- **Loss and Generalization in MobileNetV2:** Despite the slightly lower accuracy, MobileNetV2 achieved a better (lower) validation loss of 0.212 compared to ResNet50's 0.217. This suggests that while ResNet50 makes slightly more correct predictions overall, predictions made by MobileNetV2 are more confident on average. The lightweight nature of MobileNetV2, utilizing depthwise separable convolutions, may act as a natural regularizer, rendering it slightly less prone to overfitting on the relatively small Caltech-101 dataset compared to the heavier ResNet50 model.
- **Feature Extraction vs. Fine-Tuning:** Unfreezing the deeper layers allowed both models to adjust higher-order feature representations to the specific characteristics of the Caltech-101 dataset, which proved critical for achieving greater than 93% accuracy scores.

Ultimately, both models demonstrate exceptional transfer learning capabilities on this dataset. The choice between the two depends on the deployment scenario: ResNet50 is optimal for maximizing absolute accuracy, while MobileNetV2 provides nearly identical performance with greater efficiency and stability in its loss curve.

### V. CONCLUSION

This paper detailed an end-to-end transfer learning pipeline for object recognition on the Caltech-101 dataset. By comparing ResNet50 and MobileNetV2 architectures, and contrasting layer-freezing with fine-tuning strategies, the practical trade-offs in deep learning deployments were demonstrated. Fine-tuning consistently outperformed static feature extraction, highlighting the critical importance of domain adaptation. Furthermore, while ResNet50 achieved superior accuracy (94.14%), MobileNetV2 offered a highly compelling balance of speed and efficiency with a more stable validation loss (0.212). Future work could explore more advanced regularization techniques or hybrid architectures to further push the boundaries of accuracy on limited datasets.