kashif ur Rahman

September 2025

# 1 Project Report: Transfer Learning with VGG16

## 1.1 1. Introduction

This project demonstrates the use of a **pre-trained VGG16 model** as a feature extractor combined with a custom classification head.

The goal was to apply transfer learning on a built-in dataset (e.g., CIFAR-10 or MNIST/Fashion-MNIST if internet restrictions exist) and visualize essential outputs such as **training history, feature maps, confusion matrix, and performance metrics**.

VGG16 is a convolutional neural network architecture introduced by Simonyan and Zisserman (2014). It consists of 16 weight layers, originally trained on the **ImageNet dataset (1.2M images, 1000 classes)**.

## 1.2 2. Methodology

### 1.2.1 2.1 Dataset

- **Default dataset:** CIFAR-10 (10 classes, 60,000 images of 32×32 resolution).
- **Fallback dataset (offline environments):** MNIST/Fashion-MNIST, resized to 224×224×3.
- Data split: 50,000 training samples and 10,000 test samples.

### 1.2.2 2.2 Preprocessing

- Images were resized from original shape to **224×224×3** to match VGG16 input.
- Pixel values normalized using **Keras preprocess_input**.
- Labels one-hot encoded into categorical vectors.

### 1.2.3 2.3 Model Architecture

- **Base model:** Pre-trained VGG16 (`include_top=False`).
- **Layers frozen** to keep learned ImageNet weights.
- **Custom head added:**
  - Flatten layer
  - Dense(256, ReLU)
  - Dropout(0.5)
  - Dense(10, Softmax)

### 1.2.4 2.4 Training

- Optimizer: Adam
- Loss: Categorical cross-entropy
- Metrics: Accuracy
- Epochs: 3–10 (demo), batch size = 64

## 1.3 3. Results

### 1.3.1 3.1 Training Curves

- **Loss curve** shows decreasing training and validation loss.
- **Accuracy curve** shows gradual improvement, with validation accuracy stabilizing around expected performance.

### 1.3.2 3.2 Confusion Matrix

- Heatmap visualization of misclassifications across all 10 classes.
- Common confusions occur between visually similar categories (e.g., "cat" vs "dog", "truck" vs "automobile").

### 1.3.3 3.3 Classification Report

- Precision, Recall, and F1-score reported for each class.
- Macro and weighted averages give a global performance overview.

### 1.3.4 3.4 Feature Map Visualization

- Feature activations of early convolutional layers were extracted.
- Visualizations show edge detection, texture patterns, and progressively higher-level representations.

## 1.4 4. Discussion

- Transfer learning using **VGG16** provides strong baseline performance even with limited epochs.
- The model leverages pre-trained features, requiring only fine-tuning of the classification head.
- Visualization of feature maps confirms hierarchical feature extraction.
- Confusion matrix analysis highlights which classes are harder to distinguish, guiding possible improvements (e.g., data augmentation, fine-tuning deeper layers).

## 1.5 5. Conclusion

This project successfully implemented **VGG16 transfer learning** with a built-in dataset, achieving good classification performance and generating essential visualizations:

- **Training/validation curves**
- **Confusion matrix & classification report**
- **Feature maps from convolutional layers**

The approach demonstrates how pre-trained models can be adapted for new tasks with limited computational cost.

## 1.6 6. Future Work

- Extend training with more epochs for better convergence.
- Apply **Grad-CAM** for interpretability of predictions.
- Experiment with fine-tuning deeper VGG16 layers.
- Compare results with other architectures (ResNet, Inception).