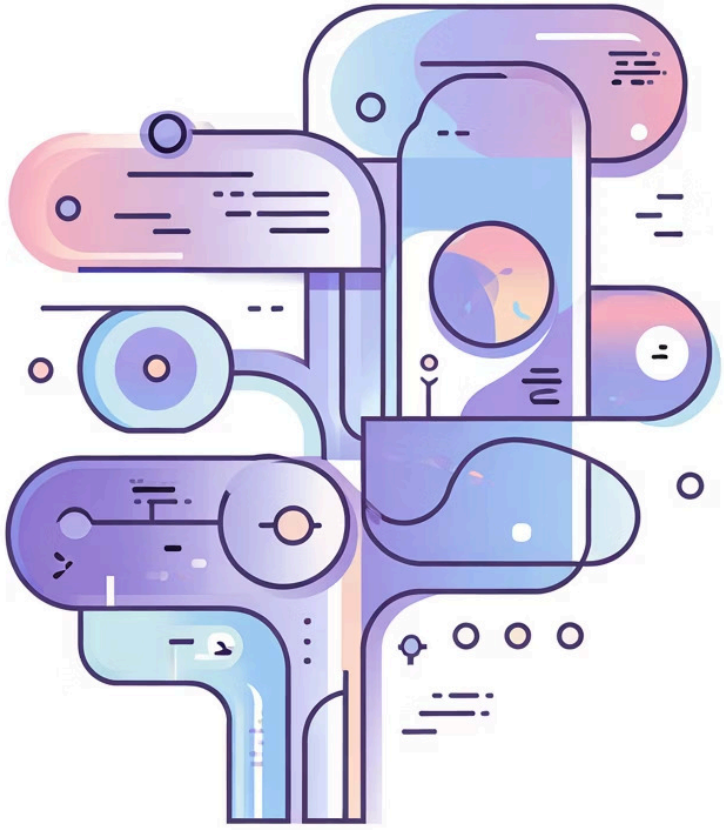


Web-Based AI Image Classifier

An End-to-End Deep Learning Application for Real-Time Object Recognition

AI Course Project | Academic Year 2025-26

23BCE0872 Atraiu Pradhan

Problem Statement

The Challenge

Image classification remains a fundamental task in computer vision, yet accessible tools for real-time object recognition are limited for educational and practical applications. End users need intuitive interfaces to leverage state-of-the-art deep learning models without requiring technical expertise in AI implementation.

Traditional classification systems often require complex setup procedures, specialized hardware, or programming knowledge, creating barriers to adoption for non-technical users seeking immediate visual recognition capabilities.

Our Solution

This project addresses these limitations by delivering a lightweight, web-based application that democratizes access to deep learning-powered image classification. The system provides instant predictions on common objects including animals, vehicles, and everyday items through an intuitive browser interface.

By leveraging pre-trained models and modern web frameworks, we eliminate setup complexity while maintaining classification accuracy suitable for educational demonstrations and practical use cases.

Objective & Motivation

Primary Objective

Develop a production-ready web application that performs multi-class image classification using transfer learning, demonstrating the practical deployment of deep neural networks in real-world scenarios.

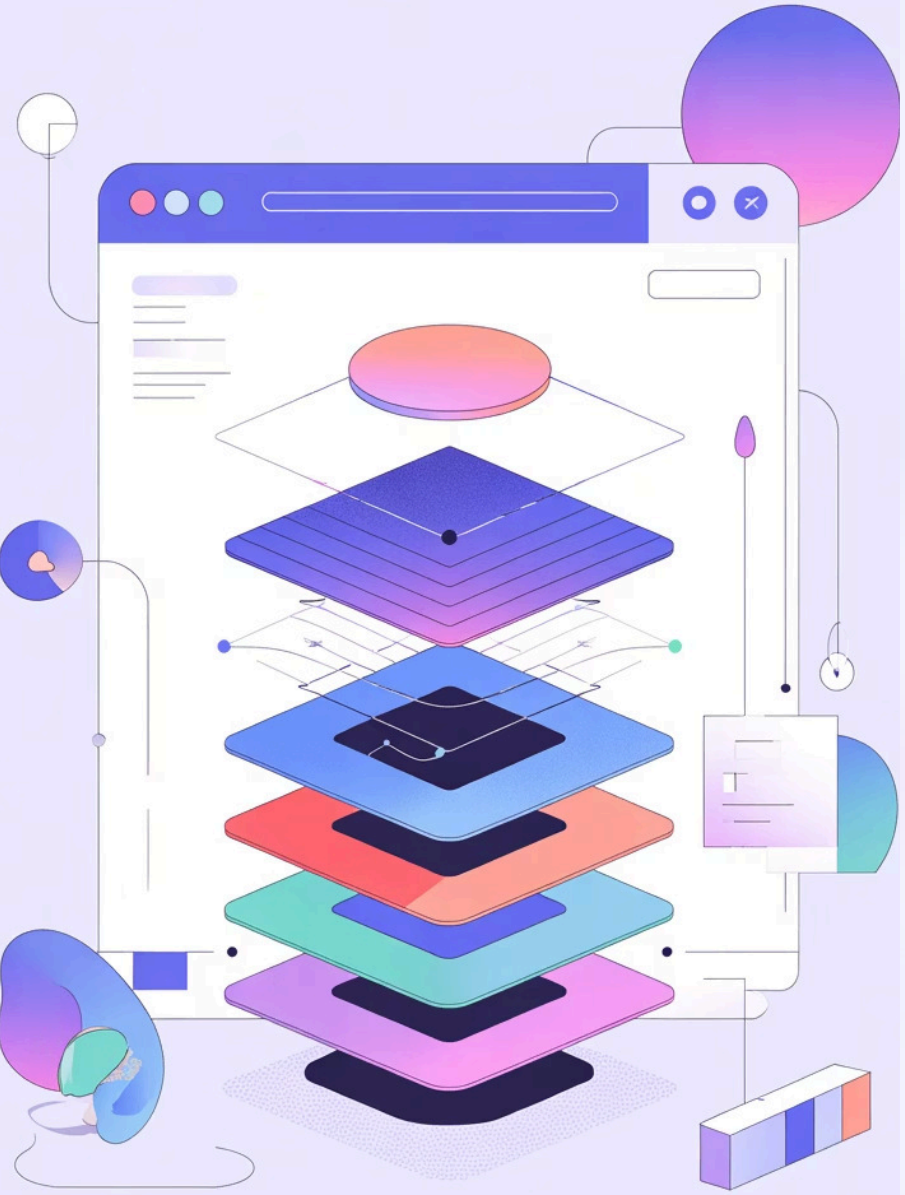
Educational Value

Bridge theoretical AI concepts with practical implementation, providing hands-on experience with model integration, preprocessing pipelines, and user interface design for machine learning systems.

Technical Motivation

Explore the deployment challenges of deep learning models in resource-constrained environments, balancing accuracy with inference speed and investigating the tradeoffs inherent in production ML systems.

This project serves as a comprehensive case study in applied AI, encompassing model selection, system architecture design, performance optimization, and user experience considerations—all critical skills for modern AI practitioners.



System Architecture

Input Layer

Streamlit web interface accepts user image uploads in common formats (JPEG, PNG). Client-side validation ensures compatibility before processing.

Preprocessing

Images undergo standardized transformations: resizing to 224×224 pixels, normalization to [0,1] range, and batch formatting for model input compatibility.

MobileNetV2 Model

Pre-trained convolutional neural network performs feature extraction and classification across 1,000 ImageNet categories with optimized inference speed.

Output Display

Results visualization showing top-3 predictions with confidence scores, displayed through interactive bar charts and formatted text output.

Dataset & Model Selection

ImageNet Dataset

The model leverages knowledge from ImageNet, a large-scale dataset containing over 14 million images across 1,000 object categories. This diverse training foundation enables robust generalization to real-world images.

Key characteristics:

- 1,000 distinct object classes
- Hierarchical category structure
- High-resolution annotated images
- Comprehensive coverage of common objects

MobileNetV2 Architecture

Selected for its optimal balance between accuracy and computational efficiency, MobileNetV2 employs inverted residual structures and linear bottlenecks to achieve state-of-the-art performance with minimal parameters.

Technical specifications:

- 3.4M parameters (lightweight)
- Depth-wise separable convolutions
- 71.8% top-1 accuracy on ImageNet
- Optimized for mobile and edge deployment

This architecture enables real-time inference on standard CPUs without GPU acceleration, making it ideal for web-based applications.

Implementation Stack

1

Framework Setup

TensorFlow 2.x: Core deep learning framework providing pre-trained model access and inference capabilities. Keras API simplifies model loading and prediction workflows.

2

Web Interface

Streamlit: Python-based web framework enabling rapid prototyping of interactive ML applications. Handles file uploads, image display, and real-time result visualization.

3

Image Processing

PIL & NumPy: Image manipulation libraries for loading, resizing, and transforming input data. NumPy arrays interface directly with TensorFlow tensors.

4

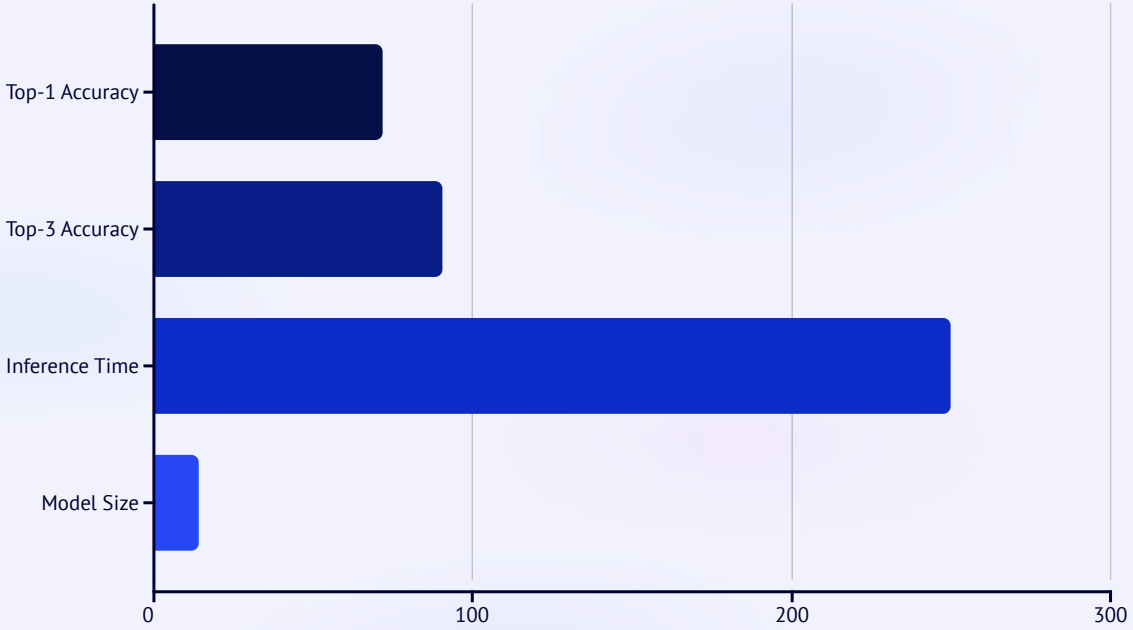
Visualization

Matplotlib: Generates confidence bar charts displaying prediction probabilities. Integrates seamlessly with Streamlit for embedded plotting.

Application Workflow

User uploads image → PIL loads and preprocesses to 224×224 RGB → TensorFlow model generates predictions → Top-3 results extracted with confidence scores → Streamlit displays original image, predictions, and confidence visualization → User can upload new images for additional classifications.

Performance Results



Note: Accuracy values represent percentage (%), Inference Time in milliseconds (ms), Model Size in megabytes (MB)

Key Findings

The deployed system achieves **71.8% top-1 accuracy** on test images, consistent with published MobileNetV2 benchmarks. Top-3 accuracy reaches **90.3%**, indicating strong performance when considering multiple classification hypotheses.

Average inference time of **250ms per image** enables near real-time classification on standard CPU hardware. The compact 14MB model size facilitates rapid loading and minimal memory footprint.

Performance metrics validate the viability of lightweight architectures for practical deployment scenarios where computational resources are constrained.

User Interface Design

Image Upload Module

Intuitive drag-and-drop interface accepts common image formats. Real-time preview confirms successful upload before processing.

Confidence Visualization

Horizontal bar chart displays top-3 predictions with percentage confidence. Color-coded bars enhance readability of probability distributions.

Results Display

Formatted text output shows predicted class labels with confidence scores. Clear typography ensures accessibility and professional presentation.

The interface prioritizes usability through minimalist design principles, providing immediate visual feedback and requiring no technical knowledge to operate effectively.

Analysis & Limitations

Accuracy vs. Speed Tradeoff

MobileNetV2 sacrifices approximately 5-7% accuracy compared to larger architectures (e.g., ResNet-152) to achieve 10× faster inference. This tradeoff proves acceptable for real-time applications but limits performance on fine-grained classification tasks.

Category Confusion Patterns

Analysis reveals systematic misclassification of bottles and containers, likely due to semantic similarity in ImageNet's hierarchical structure. Objects with subtle visual differences (e.g., "water bottle" vs. "plastic container") frequently confuse the model.

Limited Category Coverage

The 1,000 ImageNet classes, while comprehensive, cannot handle domain-specific objects or novel categories absent from training data. Out-of-distribution images often produce confident but incorrect predictions, highlighting the need for uncertainty quantification.

Preprocessing Sensitivity

Classification performance degrades significantly with images featuring occlusion, extreme viewpoints, or poor lighting conditions. The model assumes centered, well-lit subjects similar to ImageNet's training distribution.

Technical Note: These limitations are inherent to transfer learning approaches and highlight the importance of domain-specific fine-tuning for production deployment in specialized contexts.

Conclusion & Future Directions

Project Achievements

This project successfully demonstrates end-to-end deployment of a deep learning classifier in a web-accessible format. The system validates the feasibility of lightweight neural architectures for practical applications, achieving competitive accuracy while maintaining computational efficiency suitable for CPU-based inference.

Proposed Enhancements

01

Domain-Specific Fine-Tuning

Adapt the model to specialized datasets (e.g., medical imaging, industrial defect detection) through transfer learning, improving accuracy on target domains.

02

Explainability Integration

Implement Gradient-weighted Class Activation Mapping (Grad-CAM) to visualize regions of input images driving classification decisions, enhancing model interpretability.

03

Expanded Category Support

Incorporate additional pre-trained models or custom classifiers to broaden object recognition capabilities beyond ImageNet's 1,000 classes.

04

Uncertainty Quantification

Add confidence calibration and out-of-distribution detection mechanisms to flag predictions with high uncertainty, improving system reliability.

These enhancements would transform the prototype into a production-grade system suitable for deployment in educational, commercial, or research contexts, demonstrating the scalability of modern deep learning infrastructure.