COPS Summer of Code Report

M Gokulan
Roll No: 24155043
Department: Mining Engineering

Project Title:
CV Track: Convolutional Neural Networks

Abstract

This report presents the implementation and evaluation of a ResNet-50-based convolutional neural network from scratch for image classification as part of the CSOC 2025 Intelligence Guild project. The project explores transfer learning by progressively pretraining the ResNet-50 model on several datasets before fine-tuning on target datasets.The report discusses the rationale behind model selection, training methodology, dataset challenges, and the effectiveness of Grad-CAM in understanding CNN decision making.

Model Selection and Training Strategy

For this project, I chose the **ResNet-50** architecture instead of building a simpler CNN from scratch. After experimenting with multiple architectures, fine-tuning the original ResNet-50 model consistently provided better performance. This guided my decision to use ResNet-50 as the base model.

To improve the model's generalization on the target dataset, I employed a two-stage pretraining strategy:

- First, the ResNet-50 model was pretrained on the **Tiny ImageNet** dataset.
- Then, it was further pretrained on the **CIFAR-100** dataset.

The rationale behind this approach was to expose the model to diverse image distributions and object categories before fine-tuning it on the target data.

Initially, I fine-tuned the pretrained model on the **Caltech-256** dataset, but the results were unsatisfactory. Upon further experimentation, I trained the same model on the **Caltech-100** subset, which led to a significant performance boost, achieving an accuracy of **92%**.

This outcome revealed two key insights:

- The **Caltech-256** dataset has a large number of fine-grained and complex classes that make classification more challenging.
- The dataset is also **imbalanced**, which impacts the model's ability to generalize well across all classes.

This experience highlighted the importance of dataset complexity and class balance in model performance, and justified the need for progressive pretraining and careful dataset selection during fine-tuning.

Grad-CAM Visualization and Interpretability

To interpret the predictions made by the ResNet-50 model, I implemented **Gradient-weighted Class Activation Mapping (Grad-CAM)**. Grad-CAM helps visualize which regions of an input image the model is focusing on when making a decision, thereby increasing model interpretability and trust.

Why Grad-CAM?

Grad-CAM was chosen because it is architecture-agnostic, simple to implement, and effective in highlighting spatially relevant regions of interest for a given class. It works seamlessly with ResNet-50 without requiring any modification to the architecture.

Mathematical Explanation

Let A^k denote the activation map of the k-th channel in the selected convolutional layer, and let y^c be the score for class c. The weight α^k for each feature map is computed as:

$$\alpha^k = \frac{1}{Z} \sum_i \sum_j \frac{\partial y^c}{\partial A^k_{ij}}$$

where Z is the number of pixels in the feature map. The final Grad-CAM map $L^c_{\text{Grad-CAM}}$ is given by:

$$L^c_{\text{Grad-CAM}} = \text{ReLU}\left(\sum_k \alpha^k A^k\right)$$

This produces a class-specific localization map that highlights the important regions in the input image contributing to the prediction.

Results and Observations

Visualizations of the Grad-CAM outputs showed that the model often focused on the correct object regions, even when predictions were incorrect. This helped confirm that the model was learning semantically meaningful features.

- On correctly classified images, the heatmaps were well-aligned with the object.
- On misclassified or confusing images, Grad-CAM highlighted multiple or background regions, revealing ambiguity or dataset complexity.

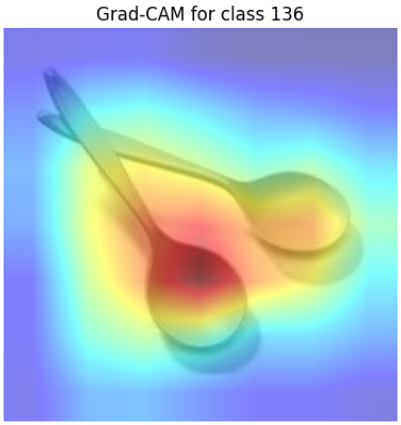

Figure 1: Grad-CAM visualization on sample images