Ayushi Sapru
COE379L
Project 4 Proposal

Exploring Neural Network Architecture Search on Clean vs Noisy Image Data

Introduction:

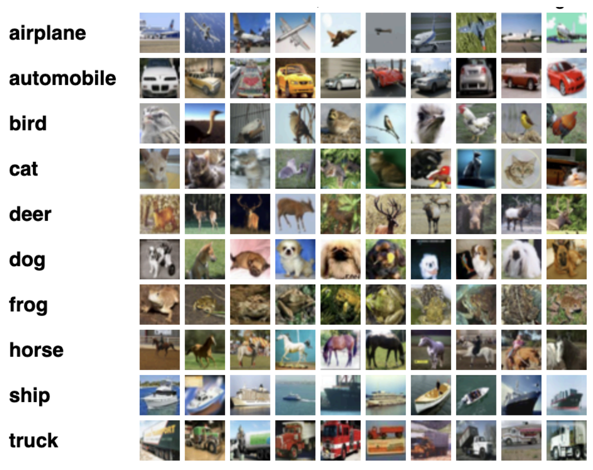
Convolutional Neural Networks (CNNs) have become a standard solution for image classification tasks. However, most model design and tuning processes assume that input images are clean and high-quality. In real-world applications, image data is often noisy as it is affected by compression, lighting, motion blur, and environmental/background interference.

This project aims to investigate how optimal CNN architectures differ when trained on clean versus noisy image data, and to understand how CNNs adapt structurally to degraded inputs. I will apply various types of noise (Gaussian, blur, JPEG, salt and pepper) to a benchmark dataset and use neural architecture search techniques to identify the best-performing models under each condition.

My main goal is to understand these concepts: how noise impacts the optimal structure of CNNs; whether certain architectural elements (like deeper layers or dropout) help models generalize better under noise; if models trained on noisy data maintain performance on clean data and vice-versa. This project is both an empirical study and a technical exploration using automated model design tools.

Data Sources:

I will use the CIFAR-10 dataset, which contains 60,000 32x32 color images across 10 classes with 6000 images per class. The dataset is well-balanced and commonly used for CNN benchmarks. To create noisy versions of the dataset, I will apply the transformations listed above. These modified datasets will be generated using Python libraries (scikit-image). All image transformations will be parameterized and reproducible.
https://www.cs.toronto.edu/~kriz/cifar.html



Methods, Techniques, and Technologies:

This project will utilize neural architecture search (NAS) to identify optimal CNN architectures for both clean and noisy image classification tasks. I will use the Keras Tuner library to define a flexible search space that includes hyperparameters like the number of convolutional layers, dropout usage, batch normalization, dense layer widths, and optimizers. The tuning process will be performed separately for each data condition - clean CIFAR-10, and CIFAR-10 with added Gaussian noise, blur, JPEG compression, and salt and pepper.

I will use TensorFlow and Keras for model development. Image transformations for noise injection will be implemented using OpenCV and scikit-image. I will log and visualize the tuning results, training metrics, and other architectural configurations using Matplotlib and Seaborn. Additionally, I will use NumPy and Pandas for data handling, and define custom metrics to evaluate accuracy and generalization across noise conditions.

Deliverables:

The primary deliverables for this project will include a set of trained CNN models, each representing the best architecture found for a specific noise condition. I will provide a comparative analysis of these architectures, and highlight the differences in structure, complexity, as well as performance. Performance metrics such as accuracy and training loss across clean and noisy datasets will be reported and visualized.

The full codebase, along with clear documentation and instructions for reproduction, will be maintained in a GitHub repository. Additionally, I will submit a final written report detailing my methodology, results, and conclusions, as well as a short video presentation summarizing the project.