**Poverty Prediction Deep Learning Model**

**ConvNN: Convolutional Nearest Neighbor for Neural Networks (Part 1)**

[**GitHub Repo**](https://github.com/mkang817415/Convolutional-Nearest-Neighbor)

Mingi Kang & Professor Jeova Farias  
Bowdoin Christenfeld Summer Research Fellowship 2024

For Bowdoin’s Christenfeld Summer Research Fellowship 2024, I have had the opportunity to work with Professor Jeova Farias on the Poverty Prediction Deep Learning Model research project.

The objective of the research project was to develop a deep learning model based on computer vision that leverages satellite imagery to estimate poverty levels in remote South American communities. This approach would have allowed us to measure poverty in regions traditionally challenging to assess.

Our proposed deep learning model that would be used for satellite imagery classification utilizes a novel convolution technique that focuses on the relationship between center pixels and their nearest/most similar neighbors, rather than just the adjacent pixels. The proposed convolution technique would be called **Convolutional Nearest Neighbor**. This approach aims to improve classification accuracy by considering a broader context within images. In our case, the model would consider neighboring structures and landmarks as points of interest within the satellite images, enhancing the traditional image recognition process.

The summer was spent implementing and creating this such model. Instead of diving straight into predicting poverty levels, we needed to implement it and test it on simple datasets to ensure that it provides some success that can later be used in the poverty prediction project.

Initially, we have built the model for classification and denoising for 1D images. The dataset we have tested was MNIST1D, a 1-dimensional representation of handwritten digits (MNIST is the original dataset of handwritten digits). After thorough implementation of the 1-dimensional model, the tests were showing promising results. Although the standalone Nearest Neighbor model was not the best performing, however we have learned that through the combination of both regular convolution and convolutional nearest neighbor was able to perform better than the standard convolutional network. This combination branching network provided solid evidence that our implemented model can learn the relationship between the center pixels and their nearest neighbors.

After initially spending time on implementing the 1-dimensional version, we have implemented the 2-dimensional version for 2-dimensional images that were greyscale and RGB images. Our implementation aimed to mimic the functionality of the traditional 2-dimensional convolution from PyTorch to which includes full customizability such as changing the number of channels, allowing pixel shuffle and unshuffled, etc. The full functionality and customizing can be found in our GitHub repository page.

For both 1-dimensional and 2-dimensional convolutional nearest neighbor process, we have created all samples, random sampling and spatially sample neighbor methods.

1. **Conv1d\_NN**: random all samples and random sampling 1-dimensional convolutional nearest neighbor
2. **Conv1d\_NN\_spatial**: spatially sample 1-dimensional convolutional nearest neighbor
3. **Conv2d\_NN**: random all samples and random sampling 2-dimensional convolutional nearest neighbor
4. **Conv2d\_NN\_spatial**: spatially sample 2-dimensional convolutional nearest neighbor

For our 2-dimensional convolutional nearest neighbor models, we tested various models for MNIST, Fashion MNIST, and CIFAR10 datasets which are frequently used in testing classification and denoising deep learning models. With promising results, we are confident that our newly implemented convolutional technique will be able to perform better than the classic convolution.

The first part of project: **ConvNN: Convolutional Nearest Neighbor** will be continuing during the fall 2024 and spring 2025. Our goal is to test more of our models and continue to make it better. Once we have created a model that can perform better/learn better, we will transition into the second part of the project: **Poverty Prediction using Convolutional Nearest Neighbor**. We aim to finish both part 1 and part 2 of the project by end of 2024-2025 school year.

For further discussion and in-depth explanation of our convolutional technique and code, please refer to our GitHub repository provided below the title.