CE 5319 Machine Learning for Civil Engineers
Exercise Set 4
Processing Images with Neural Networks

Instructions

This exercise introduces image classification using artificial neural networks. You will process grayscale images of concrete surfaces to detect the presence or absence of cracks. You will first implement a flat artificial neural network (ANN), and then a convolutional neural network (CNN), both using PyTorch. Emphasis is placed on model construction, training, experimentation, and performance interpretation.

Exercises

1. **Retrieve the Dataset.** Begin by downloading the image dataset from: *Adrien Müller, Nikos Karathanasopoulos, Christian C. Roth, Dirk Mohr, "Machine Learning Classifiers for Surface Crack Detection in Fracture Experiments," International Journal of Mechanical Sciences, Volume 209, 2021, 106698.* Code for automated retrieval is provided in the lecture notes. Alternatively, you may download and extract the archive manually into the folders `NT`, `UT`, and `ASB`.

2. **Flat ANN Classifier.** Build a flat artificial neural network (also referred to as a multilayer perceptron or MLP) using PyTorch. Your network should accept 128×128 grayscale images as input and output binary classification: `0` for "no cracks" and `1` for "cracks."

   - Train and evaluate your model separately on the images in each of the directories: NT (∼1399 images), UT (∼880 images), and ASB (∼3740 images).
   - Use a train/test split of 90% training and 10% testing.
   - Plot learning curves (loss vs. epoch).
   - Find "new" images from the internet, resize to 128×128, and test your classifier on these. At least one image should contain clear crack patterns, and one should not.
   - Experiment with different hyperparameters (e.g., `LEARNING_RATE`, number of `EPOCHS`).
   - Try different hidden layer sizes and activation functions. Switch to `ReLU` and comment on the effect.
   - Briefly summarize and interpret your model's classification performance.

3. **Convolutional Neural Network.** Build a convolutional neural network (CNN) using PyTorch. A code framework is provided in the lecture notes. As with the flat ANN:
   - Use the 128×128 images as input and predict binary class labels (`0` or `1`).
   - Process each dataset: NT, UT, and ASB.
   - Use a 90%/10% train/test split.
   - Plot learning curves (loss vs. epoch).
   - Test the model with "new" external images.
   - Experiment with different kernel sizes (e.g., 5×5 vs. stacked 3×3), hidden layer sizes, activation functions (e.g., try switching to `sigmoid`), and other hyperparameters.
   - Discuss differences in training behavior and classification accuracy compared to the flat ANN.
4. **Conceptual Questions.**
   (a) Explain why a convolutional neural network is often more effective than a flat neural network for image classification tasks. What kinds of spatial features can a CNN detect that an MLP cannot?
   (b) Suppose your ANN performs well on the training data but poorly on new images. What might this indicate about your model? What techniques could help mitigate this behavior?