MSAI-437: Deep Learning
Group Project #2 - Computer Vision
(25.0 points)

Winter 2023

Description: CCHS is a rare genetic disorder that can result in severe breathing problems. Global cases are estimated to be on the order of a few thousand. CCHS is most commonly diagnosed with a genetic test, which can be prohibitively expensive. Medical professionals seek a simple and inexpensive screening tool to broadly identify high-probability cases before administering a more expensive genetic test https://www.nature.com/articles/pr200610.pdf).

The underlying genetic disorder often presents a phenotype in distinct facial features. Experts in this field can identify potential cases based on these facial features. However, it is not tractable to do so at scale. In this assignment, you will implement binary image classifiers on a low-resource dataset. In place of children's faces, you'll work with a dataset of approximately 1,300 images of the leaves of healthy and unhealthy bean plants (the "Beans" dataset, see https://github.com/AI-Lab-Makerere/ibean/blob/master/README.md). This proxy task should provide helpful insights into CCHS screening. You will be asked to submit your classifier's results on a blind dataset. Note: It may be challenging to obtain meaningful results on the Beans dataset due to the limited number of observations.

Tasks:

1. **Convolutional Neural Network** (10.0 pts): Implement a Convolutional Neural Network to perform binary classification on the Beans dataset. There are a large number of examples available on the Web. The size, design, complexity, etc., of your Convolutional Neural Network, is up to you. You are permitted to consult these, but please implement your own code. Train your model on the Beans dataset and report results on the validation set as a confusion matrix with labeled axes. In addition, use your Convolutional Neural Network to label images in the blind test set, which consists of about 40 unlabeled images.

   Please describe the operation of your model and hyper-parameter settings. Present the learning curves for training and test sets. Please discuss any limitations of your approach and possible solutions. Please do not exceed one page for this.

2. **Autoencoder** (10.0 pts): Autoencoders reproduce images using learned latent representations of input images. Since the objective function of an autoencoder is to generate an exact copy of the input image, they can be trained at scale from unlabeled images. This may be useful to a CCHS screen model in that many images are available for individuals without CCHS. More specifically, using an autoencoder trained to reproduce images of patients without CCHS is likely to struggle to encode and decode an image of a patient with CCHS. Using a large dataset of healthy leaves to train an autoencoder may provide valuable artifacts when asked to encode and decode images from the Bean dataset.

   Build an autoencoder using an architecture of your choosing to implement this approach, and train it on healthy leaf dataset(s) of your choosing (see https://www.kaggle.com/datasets/csafrit2/plant-leaves-for-image-classification). You should use the Conv2d() and ConvTranspose2d() PyTorch functions discussed in class. Then use this autoencoder to perform classification on the Bean dataset. The classification methodology is up to you. You can use the latent encoding of images from the Bean dataset directly, measure the decoded image's error relative to the input image, or use any other method you choose. Report results on the validation set as a confusion matrix with labeled axes. In addition,

apply your methodology to label images in the blind test set, which consists of about 40 unlabeled images.

Please describe the operation of your model and hyper-parameter settings. Describe your classification methodology. Please discuss any limitations of your approach and possible solutions. Please stay within two pages for this.

3. **Other Methods** (5.0 pts): Please discuss (but do not implement) different approaches that might effectively classify images with a small number of labeled samples. Include a rationale for why these approaches might be effective, sample architectures and a description of evaluation metrics. Please do not exceed one page for this.

What to Submit: Please submit a single `.zip` file for the group. The zip file should contain the following:

- A single `.pdf` for all written responses, tables and graphs,
- A single `.py` file will all code for the Convolutional Neural Network and Autoencoder (note: please do not submit Jupyter Notebooks),

Only one group member needs to make a submission.