Topic: Sign Alphabet Recognition

Chosen dataset

We chose this dataset (https://www.kaggle.com/grassknoted/asl-alphabet) because it was said to be curated for convolution neural networks, and we see on the rubric that it's a topic that we will cover in week 8 lecture. We felt that this would be a good way to apply and practice the knowledge we obtained from the lecture. The topic (sign language recognition) is also one that we are very interested in. In addition, we planned to generate variant datasets in the following ways: rotation, flipping, resizing, cropping, and normalizing of images.

Methodology

Data Preprocessing

The dataset is very feasible. Since 3000 images of hand symbols are provided for each alphabet, this will ensure that the model has enough data points. The images are taken at a variety of angles, skin color of hands, and background color, which provides good variation in the dataset.

The most useful information is the hand and its position. Some preprocess procedures are centering image data by subtracting mean pixel values from each pixel and using python.keras module to transform images into columns and rows.

Machine Learning model

The dataset is expected to predict the corresponding alphabet of the input sign image. The model we would like to adopt is a convoluted neural network(CNN). These models are very good at picking out patterns in images and has little dependence on pre-processing. These networks are multi-layered, and as we go down the layers can recognize more complex shapes. These layers have 'filters' (convolutional kernels) to detect the image patterns. We can think of it as sliding the filters over the image matrix and taking the dot product (as a means to apply weight and biases to the input pixel) to detect particular features (edges, corners, etc.). There are various architectures, such as AlexNet, LetNet-5, etc that avoid the need for feature extraction. In addition, we plan to use the softmax function as the probability function in identifying the class.

In comparison to support vector machines(SVM), they can both approximate linear and nonlinear boundaries in classification problems. SVM requires more prepossessing procedures than CNN and if the number of class labels is too large, it could be problematic for SVM. An algorithm that uses pixel vectors loses spatial interaction between pixels, a CNN uses adjacent pixel information to effectively downsample the image input.

Evaluation Metric

We will use a confusion matrix and accuracy score to determine the performance of our algorithm, the expected accuracy is 95%.

Application

User Input and Output:

The user is prompted to upload an image that contains a sign alphabet with a contrastive background to the skin color. The output will be displaying the corresponding alphabet to the screen.

Reference:

https://analyticsindiamag.com/hands-on-guide-to-sign-language-classification-using-cnn/%20:%20hands%20on%20guide

https://machinelearningmastery.com/best-practices-for-preparing-and-augmenting-image-data-for-convolutional-neural-networks/

https://deepai.org/machine-learning-glossary-and-terms/convolutional-neural-network

https://www.baeldung.com/cs/svm-vs-neural-network

https://vitalflux.com/accuracy-precision-recall-f1-score-python-example/

https://analyticsindiamag.com/why-convolutional-neural-networks-are-the-go-to-models-in-deep-learning/