Lab 2 - Feature Generation, Feature Extraction & Feature Selection

-Anthony Allen, Parth Patel, Caijun Quin

This lab focuses on initializing a machine learning algorithm to read sign language.The data used to train and test the software is collected from the class of EEL4930 and includes 1844 samples of 100x100 RGB images. These images include alphabetic sign language gestures ranging from letters A-I. This includes 205 images per letter except or letter “I” which only has 204. The dataset is further expanded by including images from a Kaggle mnist dataset. This data is analyzed and truncated to ensure it complements the existing dataset. The group decided to expand the existing data to increase the accuracy of the machine learning model when applied to test data at the end of the semester.

While the Mnist dataset provides more images to train over, the class dataset the model will be tested against at the end of the semester includes lower quality images. This is likely the case because the training set supplied by the class has some fundamental issues. Many of the images were taken at an angle, offering a nice skewing variable to be accounted for. The majority of the supplied images were cropped. However, a small percentage of them include the entire arm of the person signing. Many students collected images against a background of a similar color to their skin which may make segmentation difficult. Additionally, many of the students collected images atop a marble background which introduces a “salt and pepper” issue. Luckily, it appears that all images were taken with the right hand. The machine learning model will likely need to include a cropping algorithm and potentially an averaging filter to smooth out the noise.

From the class dataset, various normalization techniques were used. Normalization is important because it reduces all data to the same scale. With normalized data, all components can be analyzed equally without large variables overshadowing smaller ones. The normalization techniques considered for this dataset are as follows.

constancy where objects are assigned a relatively constant color regardless of illumination.

1. RGB Color channels normalization: This technique implemented the following equation

$$\text{New image} = \text{old image} / S$$

Where

$$S = \text{red} + \text{green} + \text{blue}$$

In this approach the original RGB image is separated into three color channels (red, green and blue). The intensity of each pixel is added across all channels element-wise to produce the matrix S. This matrix is then divided into the original image “old image”

elementwise to produce the new color normalized image. This approach is particularly useful for object recognition. The idea is to offer a sense of color

2. Pixel normalization :

    Xtrain = Xtrain/255

    RGB channels were transformed into grayscale and their intensity was reduced by a factor of 255. This provided intensity values between 0 and 1 and condensed the data to a single channel. The advantage this approach offers is a more manageable dataset. Small values won't be as easily overshadowed by larger values. This helps make any optimization processes numerically stable. The only disadvantage is a reduction in sparsity.

3. Min-Max scaling :

    Xtrain_technique_1 = (Xtrain - Xtrain.min()) / (Xtrain.max() - Xtrain.min())

    Min-max scaling offers a similar numeric reduction as pixel normalization with increased sparsity. Instead of limiting the 0 to 255-pixel intensity ranges to values between 0 and 1, the range is reduced and spread across a range between -1 and 1. This gives a mean offset of 0. However, this technique is most useful when normalized values have a gaussian distribution.

4. Standard Scaler approach

    This approach implements the same technique as the min-max scaling normalizer. However, it ensures that the mean value is in fact 0. This is particularly useful when the the pixel values do not have a gaussian distribution.

The next step of analyzing the class dataset involves generating features from the training set and assessing their quality. This is done by computing a set of cluster validity-type metrics.
The feature generators used were Principal Component Analysis (PCA) and Histogram of Oriented Gradients (HOG). Upon applying PCA to the original dataset, it’s seen that the cumulative explained variance exceeds 90% when approximately 600 components are used. Upon applying the HOG approach, it is seen that the features are sensitive to object orientation. This is because the spatial region being analyzed is small. To ensure the class cluster were far away from each other, the following metrics (as defined by Catia Silva) were computed.

1. Calinski-Harabasz Index: The index is the ratio of the sum of between-clusters dispersion and of inter-cluster dispersion for all clusters (where dispersion is defined as the sum of distances squared).
2. Davies-Bouldin Index: This index computes the average similarity between clusters, where the similarity is a measure that compares the distance between clusters with the size of the clusters themselves.
3. Silhouette Index: The silhouette index measures how well class group cluster together it estimates the average distance between clusters.
4. Pairwise Euclidean Distance: The pairwise Euclidean distance measures how far away class mean are from each other.

TABLE I

Calinski-Harabasz Index (Hog)	10.313954793295377
Davies-Bouldin Index (hog)	8.919833495048156
Silhouette Index (hog)	0.0051481808602665945
Calinski-Harabasz Index (PCA)	3.559216044810615
Davies-Bouldin Index (PCA)	13.179788145154127
Silhouette Index (PCA)	-0.010288608704761837

Calinski helps choose the proper clustering. From table I, we see that the index for the HOG method is much higher than that of the PCA method. Due to this fact we see that the HOG method supplies a better solution in this case. Furthermore, for the Davies-bouldin index, we see that the value for HOG is lower which again results in a relatively better clustering. Lastly, the Silhouette index from HOG has the closest value to 1. Therefore, the HOG approach is best in this case. Overall, we see that that HOG approach produces the best cluster over all forms of analysis.