Practical Machine Learning

Maria-Simona Cioclov

Project 2 - Unsupervised Methods

Contents

1 Dataset

The dataset consists of 15000 colored images of 300x200 pixels each. The images are organised in three classes of 5000 samples each, representing hand gestures from the Rock-Paper-Scissors game. The task is to predict which of the three gestures - Rock, Paper, or Scissors - the hand in an image illustrates.

I preprocessed the dataset by first converting each sample to a grayscale image while keeping the original size of the images, as illustrated in Figure 1. Additional preprocessing steps were applied separately for each feature extraction method, as I will present next. Finally, I split the dataset into training and test parts using a ratio of 80% and 20% of the whole dataset respectively. In both subsets, the classes are distributed equally.

2 Features

2.1 HOG features

One type of feature I used for classifying the hand gestures was extracted using the Histogram of Oriented Gradients method, which is a popular technique for detecting objects in computer vision. It works by splitting the grayscale image into smaller cells and computing the gradient of each pixel inside them. For each cell, a histogram of the orientations of the corresponding pixel gradients is created. These histograms are then normalized in groups of cells called blocks, to reduce the influence of lighting or contrast variations. In the end, the feature vector for a sample is formed by concatenating all the histograms of oriented gradients.

I experimented with different parameters for creating the features while plotting the points in an interactive 3D plot, such that I can visualize how well the features separate the different clusters. I used the t-SNE method to reduce the dimensionality of the dataset before plotting the clusters as it can be seen in Figure 2. The best cluster separation was achieved using this configuration: cells of 32x32 pixels, blocks of 4x4 cells, 10 orientation bins per histogram and Euclidean distance for normalizing the blocks. With this setup, each image is now described by 2880 features. To reduce dimensionality, I applied PCA while keeping 95% of the variance, which resulted in a final feature vector of 144 features per sample.

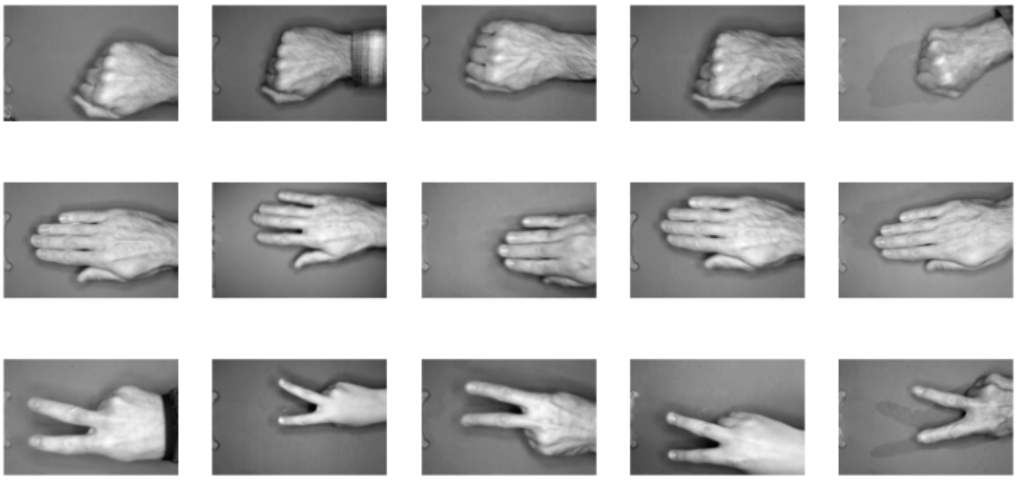

Figure 1: Example grayscale images from the dataset. Each row represents one of the three classes: Rock, Paper, and Scissors.

Figure 2: Labels represent: 0 - rock; 1 - paper; 2 - scissors

2.2 Sobel features

Another type of feature used for classifying the hand gestures was extracted using the Sobel operator, which detects edges in an image, as illustrated in Figure 3. This method computes the gradients of pixels in vertical and horizontal directions, detecting regions with high changes in intensity. For each direction, a specific convolution filter is applied to generate gradient maps. In the end, the two gradient maps are combined to compute the orientation and magnitude of each pixel. The feature vector is formed by concatenating the pixel magnitudes.The best cluster separation was achieved using 7x7 pixel kernels, with Gaussian blur applied to the image beforehand to reduce noise. With this setup, each image is now described by 60000 Features. To reduce dimensionality, I applied PCA while keeping 90% of the variance, which resulted in a final feature vector of 638 features per sample.

A representation of data represented by Sobel features can be seen in Figure 4 where I used t-SNE method to plot the data in a 3D space.

3 Birch (Balanced Iterative Reducing and Clustering using Hierarchies)

3.1 Algorithm

BIRCH is a clustering algorithm that works by incrementally building a Clustering Feature Tree (CF) to efficiently summarize the dataset. Each node in the tree represents a cluster summary, storing the number of points, the sum of their coordinates, and the sum of squared coordinates. Tree nodes are grouped in clusters based on a threshold hat determines how close points must be to be merged. If the tree becomes too large, similar clusters can be compressed to reduce complexity. BIRCH will find the best way to merge the leaf nodes until it reaches the desired number of clusters

The advantages of BIRCH is that is that points can be added in clusters without retraining from scratch. It is also low memory usage, as it compresses data into compact structures. However, it is sensitive to distance metrics, so it struggles with detecting complex cluster shapes.

3.2 Hyperparameter tuning

For tuning the hyperparameters I used the grid search technique that evaluates all possible combinations of specified hyperparameter values, as shown in Table 1. For validating the results I used cross-validation using 5 folds. This method ensures that the model's performance is evaluated on multiple subsets of data, revealing its generalization capability and reducing the risk of overfitting.

After training using HOG features, the best parameters found are: `n_clusters=3`, `threshold=2.0`, `branching_factor=20`, with a silhouette score of **0.1302**.

After training using Sobel features, the best parameters found are: `n_clusters=3`, `threshold=0.1`, `branching_factor=10`, with a silhouette score of **0.0277**.

3.3 Results

In Figure 5 and Figure 6, we can see a comparison between the original clusters and the predicted clusters for the test dataset. The predictions were made using the best model found through grid search after training and validating with HOG feature samples, in the first figure, and Sobel features, in the second figure. In both figures we can observe that the Rock label is well identified, while the Paper and Fork samples are a little bit confused.

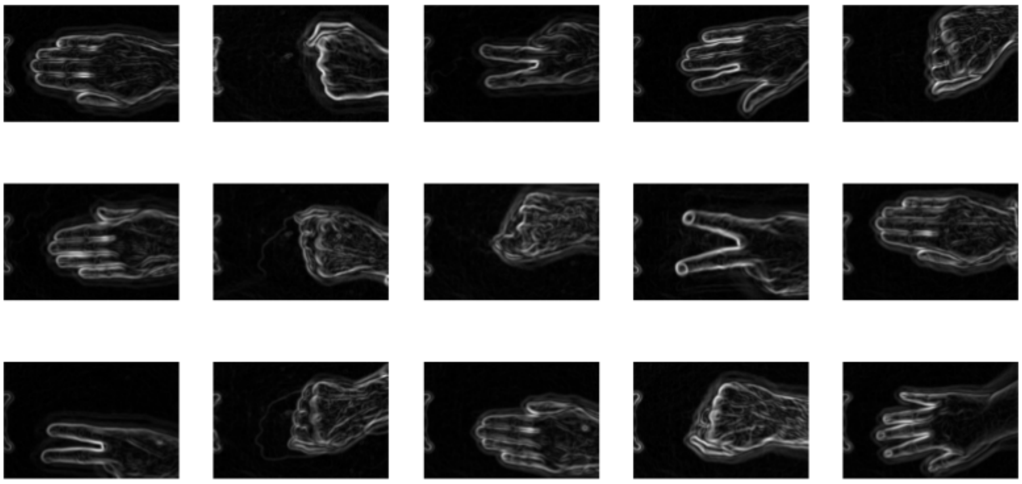

Figure 3: Example images after applying the Sobel filter.

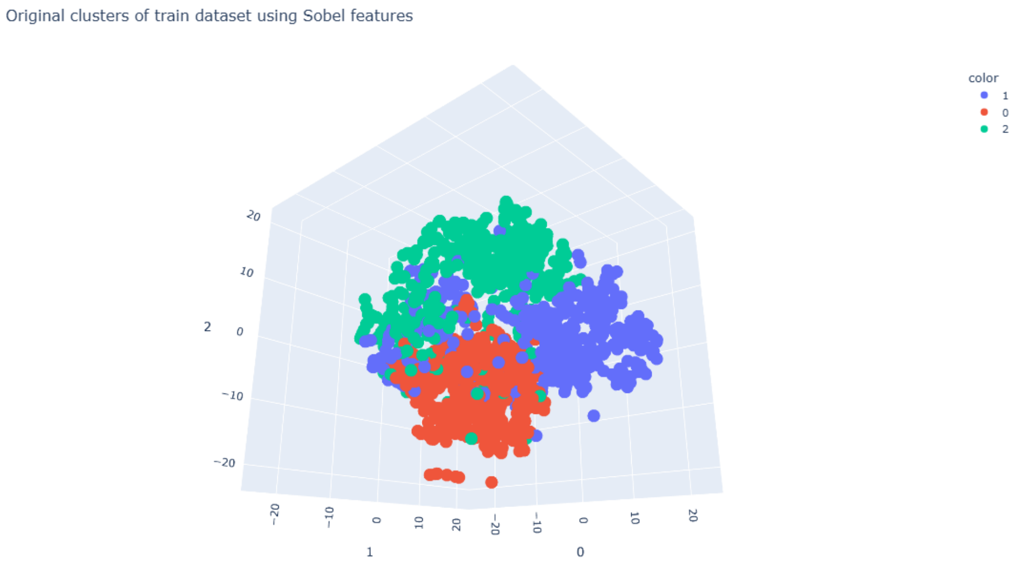

Figure 4: Labels represent: 0 - rock; 1 - paper; 2 - scissors

Parameter	Values Tried	Description
`n_clusters`	3	number of clusters to form
`threshold`	0.1, 0.5, 1.0, 2.0	distance threshold for clustering subclusters
`branching_factor`	10, 20, 50	maximum number of subclusters in each node

Table 1: Parameters tuned and values tried for BIRCH clustering

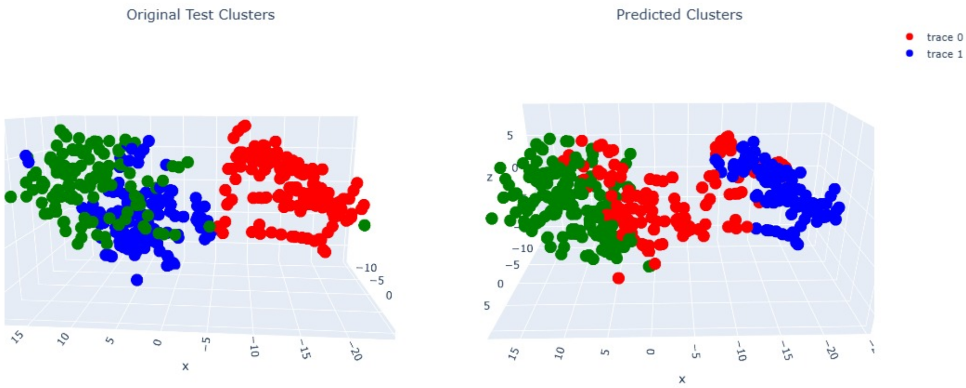

Figure 5: Comparison between the original clusters and the predicted clusters for the test dataset with HOG features.

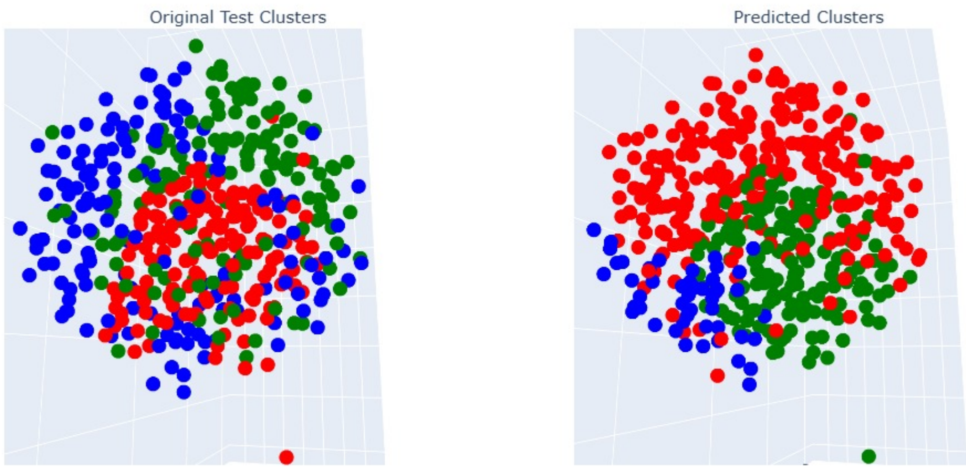

Figure 6: Comparison between the original clusters and the predicted clusters for the test dataset with Sobel features.

The random chance approach gave a Silhouette score of -0.0025 for HOG features and -0.0019 for Sobel features, both scores being lower than the ones achieved by the Birch models.

I also compared the results with two supervised methods, AdaBoost and Extra Trees. These models resulted in higher Silhouette scores, with AdaBoost scoring 0.83 and Extra Trees scoring 0.95, performing considerably better than the unsupervised method.