Project 2 overdue

Bouruc Petru-Liviu

31/08/2023

Contents

1 Introduction & Project Description

This project was about applying unsupervised methods on dataset and observe the results. For this one, I chose as a dataset english handwritten characters. The dataset contains 3410 images partitioned into 62 classes. For the supervised methods, I chose AgglomerativeClustering and Mean-Shift. The data is not separated, so we will use the build-in function *train_test_split* from sklearn to make our datasets.

2 Data Preprocessing

As for the data preprocessing, I chose to go with 2 variants: simple HSV features from image pixels, and features obtained from getting the image through a VGG network.

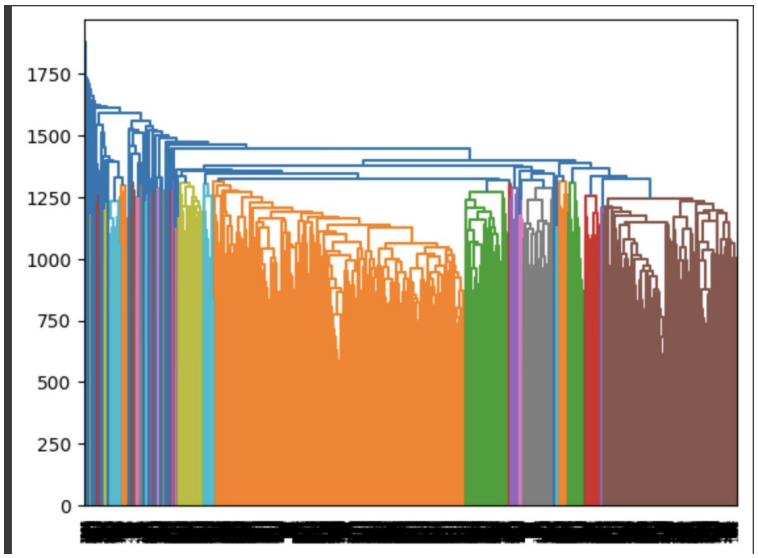
The HSV model is the most accurate representation of how humans perceive color. More than that, after converting to HSV I took the mean value of the pixels and flatten the image in order to have less computations to perform.

As for the second approach, firstly I read the images and simply saved them as arrays. Then I put them through a standard VGG pre-trained model with weights from imagenet, also making use of the build-in keras method *preprocess_input*. For this approach I also tried to use PCA to reduce the dimension of my input and also keep the core information.

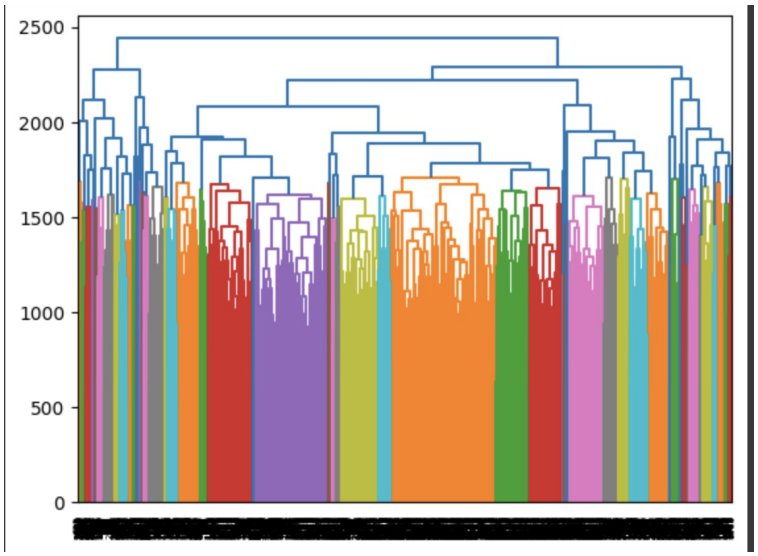
3 Agglomerative

Agglomerative Clustering is a type of clustering in which objects that are close to each other are placed into groups. It works similarly with a tree, such that at the beginning, all objects are individual clusters (leaves) and the algorithm builds the tree from the bottom to the top (root). The similarities between clusters can be computed with different methods, a combination between metrics and linkages. This structure is called a dendrogram, being a common type of hierarchical clustering.

Dendogram representation of our dataset with VGG features and reduced with PCA, with average linkage:

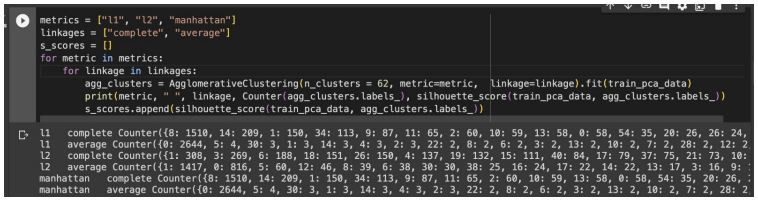


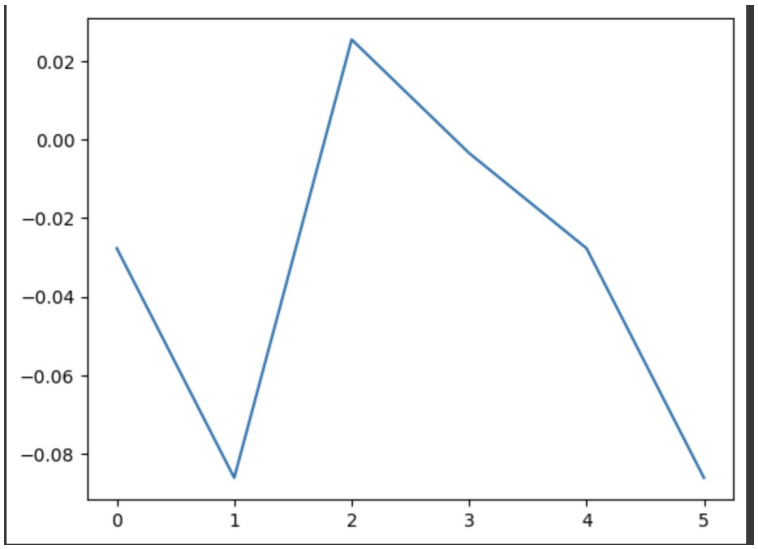
Dendogram representation of our dataset with VGG features and reduced with PCA, with complete linkage:



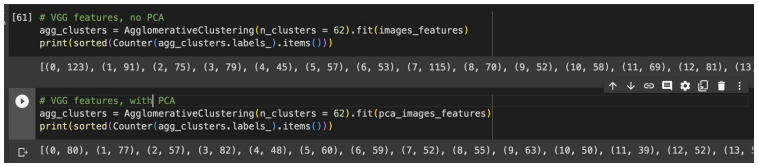
When computing the best combination for Agglomerative Clustering, I took into account

different metrics and linkages. As it can be seen, none of them gave exceptionally results by looking at the silhouettes score.





One last experiment which I tried with agglomerative was to clusterize all the dataset with and without PCA applied, in order to see if there is a big difference between them. As we can observe, the clusters are approximately the same sizes, which could be an indicator that after the transformation much of the information was being kept.

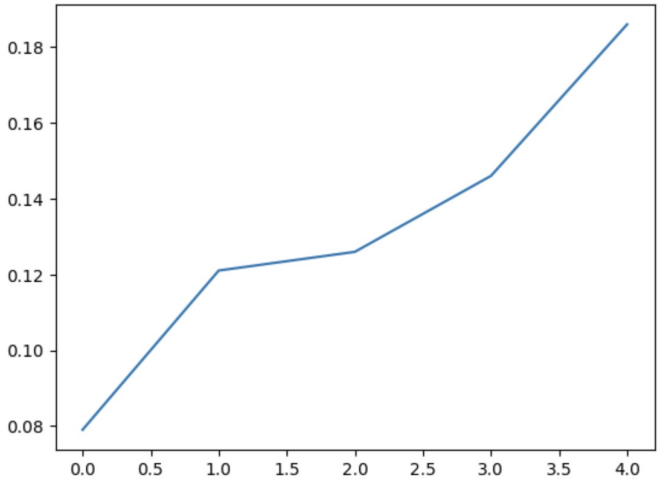


4 Mean-Shift

The second approach was Mean-Shift, which is a clustering algorithm that is based on the idea that it finds high density in a region and move the other points towards that. The shift is done with the help of a kernel, in which the distance of which the mean is calculated is represented by the bandwidth. Sklearn offers us a build-in nearest-neighbor analysis with the help of *estimate_bandwidth* function. Using this, on my dataset it approximated a bandwidth of around 1200, so I computed the Mean-Shift with values from that range.

It was a heavily computational task, which took some time to complete. In the end, an improvement could be seen on terms of silhouette scores.

Using the HSV features, the Mean-Shift algorithm could not make the clustering in reasonable time, as the number of features proved to be too big and it's runtime exceeding one hour. So, for this approach, I could only use the second set of features, those based on VGG and then reduced with PCA.

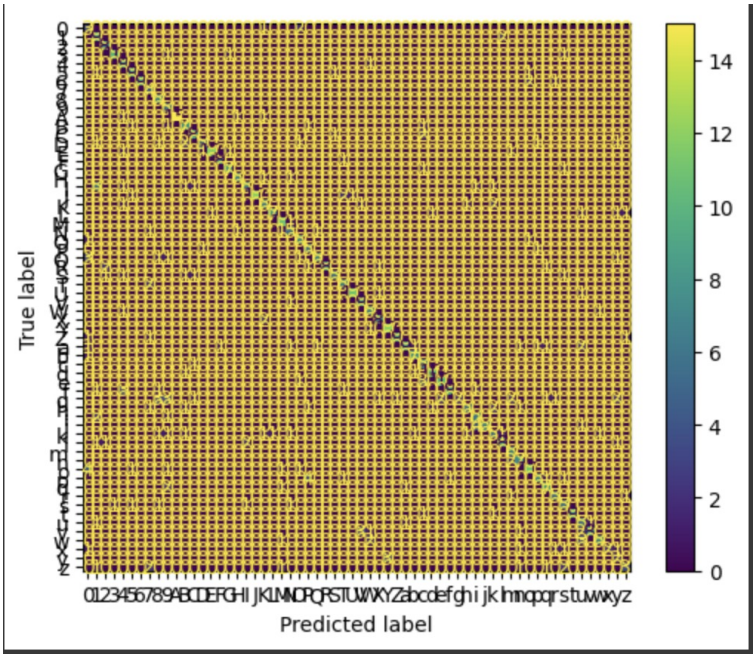


5 Comparison with supervised method and random chance

For the supervised method, I chose to work with RandomForest classifier, and as for the dataset, I used the VGG-features after PCA transformation. I tried a grid search to find the best parameters in order to maximize the accuracy.

The results were:

	n=100	n=200	n=500	n=100
kernel=rbf	0.7	0.74	0.77	0.78
kernel=poly	0.64	0.69	0.76	0.76



As for the random chance, by assigning random labels to each data point from the possible values of the dataset labels, the accuracy that it managed to obtain was 0.017. It is so low because of the great number of classes.

6 Conclusions

In conclusion, it is clear that the supervised methods outperform the unsupervised one for this task, but with the right improvements, they can give a good analysis of the dataset.