Neural Network Clustering for Milan’s Aerial View

Introduction

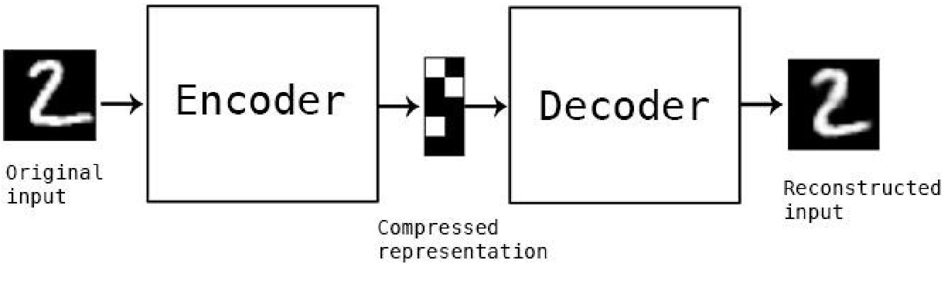
The general goal of this project is to analyze and cluster several sections of Milan's aerial view. This work is part of a project of departments DEIB and ABC of Politecnico di Milano. In particular, this project is intended to explore the clustering of images representing Milan’s aerial view using a neural network to perform feature extraction. The final goal is to understand how different areas of the city are clustered together and what are the similarities leveraged to perform such clustering.

Feature Extraction

Neural networks have been used to perform the feature extraction part since they are able to learn complex non-linear representations of data. For this purpose, most of the approaches leverage neural networks to provide a compact representation of data. Subsequently, the compact representation is fed to a clustering algorithm that groups data points together. The dataset available is obtained by cutting the aerial view of Milan into a grid of different edges’ measures: 200,400 and 800 meters. The dataset having the edges’ size of 200 meters has been chosen to carry out the above mentioned analysis. In particular, it contained 3883 black and white images representing different areas of the city. The higher the edge’s measure, the lower the number of images in the dataset; choosing the size of 200 allowed to have a higher number of images, despite being still relatively low for the training of a neural network.

Autoencoders

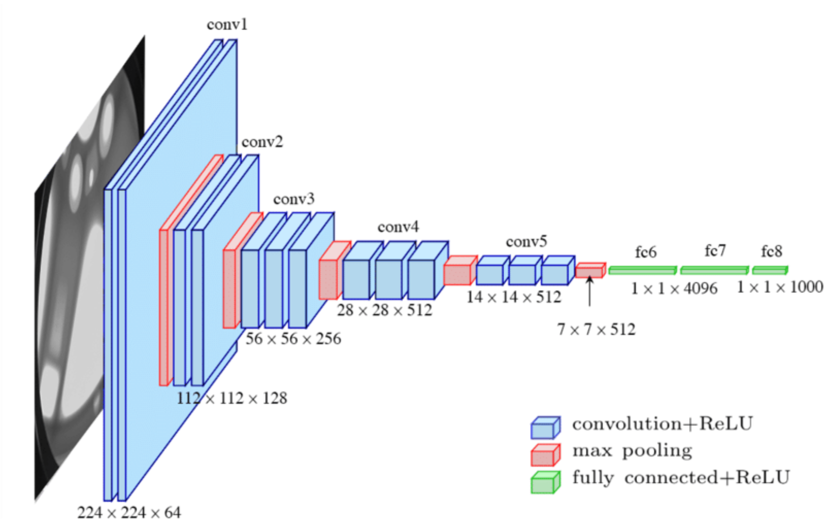
The first approach to carry out this task is through the use of autoencoders. They are made of two parts: **encoder** and **decoder.** Basically, the training of the network is based on the minimization of the reconstruction error. Hence, given an input image, it is compressed in a sparse representation at the end of the encoder and it is reconstructed during the decoding phase. Therefore, the reconstruction error represents how well the network is able to reconstruct images starting from a compressed representation. Having an autoencoder that performs well in the reconstruction means that its compact representation is able to express the information despite its low dimensionality.



Auto Encoders are tight to the specific compression task, therefore they need to be trained on a specific dataset, with many images. Although the choice of the edge led to the highest possible number of images, it was still relatively low for the training of the autoencoder. For this reason, the reconstruction error was high and the visual reconstruction of images seemed to be not very accurate. When the dataset is limited in size, transfer learning may represent a solution since the network is already trained over millions of images.

Convolutional Neural Networks

Convolutional neural networks perform well in extracting features from images. For this reason they have been used in this task. To tackle the problem of the relatively small dataset a pre-trained VGG16 network has been used to perform the feature extraction.



The feature extraction process provided a compact representation of the images' features. Usually, the fully connected layers are trained in order to find patterns that are specific to the current task. However, as happened in the autoencoder, the dataset was too small in order to properly train the last layers. Hence the general pattern extraction of the VGG16 has been kept for the task.

Deep Embedded Clustering

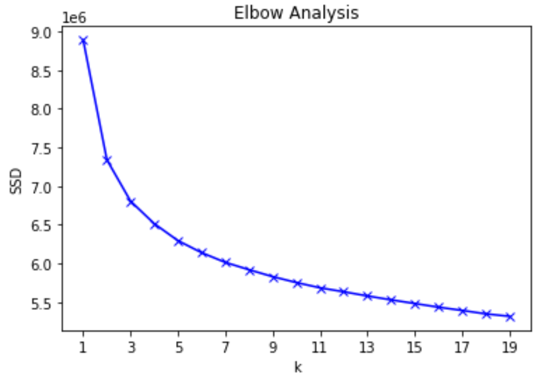
Once the compact representation of the images is obtained, clustering algorithms can be applied in order to group together images having similar patterns. This procedure is called Deep Embedded Clustering. In this work, three different clustering algorithms were experimented: **K-Means**,
K- Medoids and **Hierarchical Clustering.**

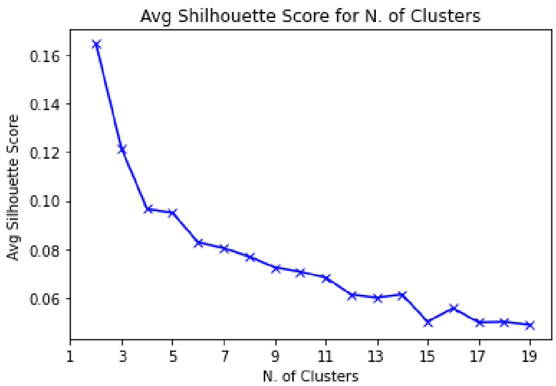
In all the three proposed algorithms, Elbow and Silhouette score analysis have been used to understand which was the best number of clusters. Regarding the elbow analysis, all the "elbows" are considered to be good values for K since they represent a significant decrease in the squared distance. The silhouette score has a range in [-1,+1] and values close to 1 means that points in clusters are close to each other and far from other clusters. Therefore, a high

silhouette score indicates good clustering, a score close to -1 indicates that the clustering results are poor, and values close to zero usually indicate overlapping clusters.
Reference : [https://dzone.com/articles/kmeans-silhouette-score-explained-with-python-exam]

K - Means

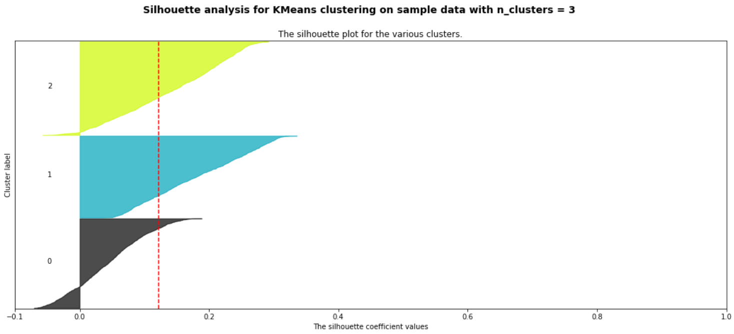
K-Means algorithm assumes that the number of clusters is defined at the beginning. Its goal is to assign data to the closest centroid. Centroids are randomly initialized and subsequently updated at each iteration.

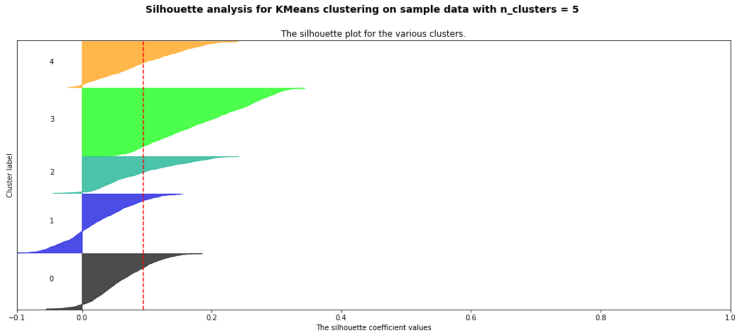


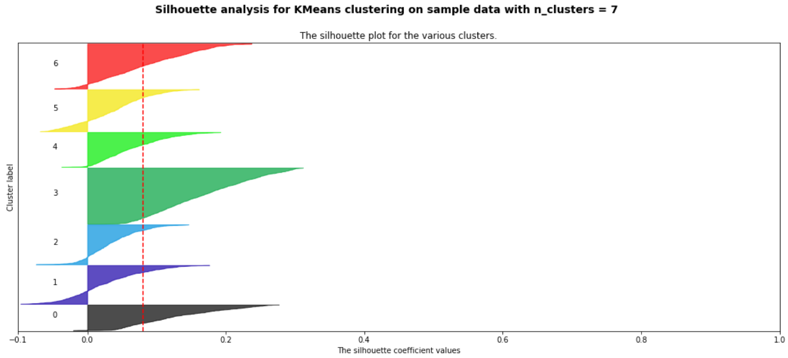


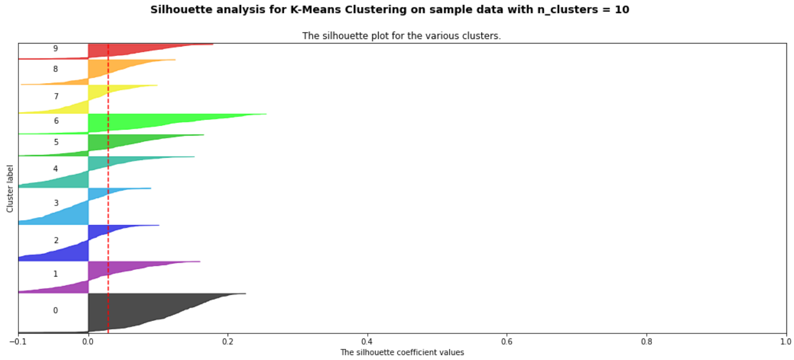
In this case, due to the smoothness of the elbow analysis, it was hard to choose a good set of k values. Therefore, the ones coming from the architecture literature have been chosen (k = [5, 7, 10]).

Silhouette Analysis





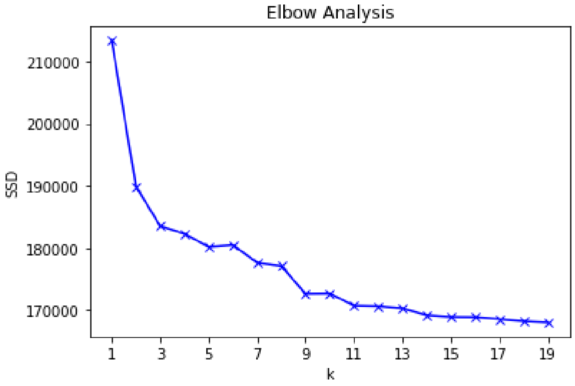


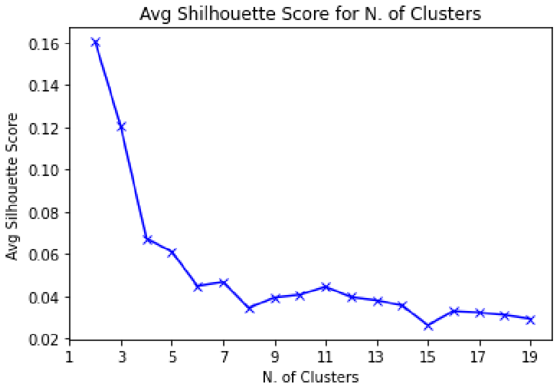


In the plots above the colored areas are made of all the points of the clusters, for different values of k. In particular, the area on the right represents the number of points with a positive silhouette score (well clustered), while the ones on the left seem to be far from points in the same cluster. It is possible to see that, according to the silhouette analysis, lower k values lead to better clustering results.

K-Medoids

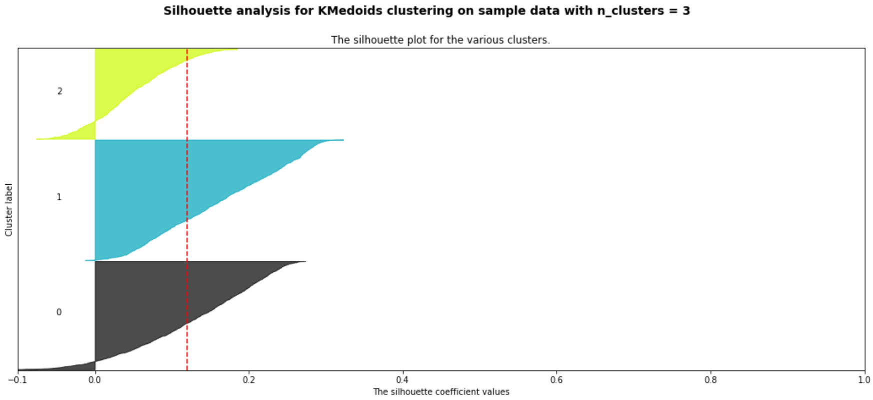
This algorithm is a variation of the Kmeans. *K*-medoids minimizes the sum of dissimilarities between points labeled to be in a cluster and a point designated as the center of that cluster. In this case, centroids are chosen among the data points.

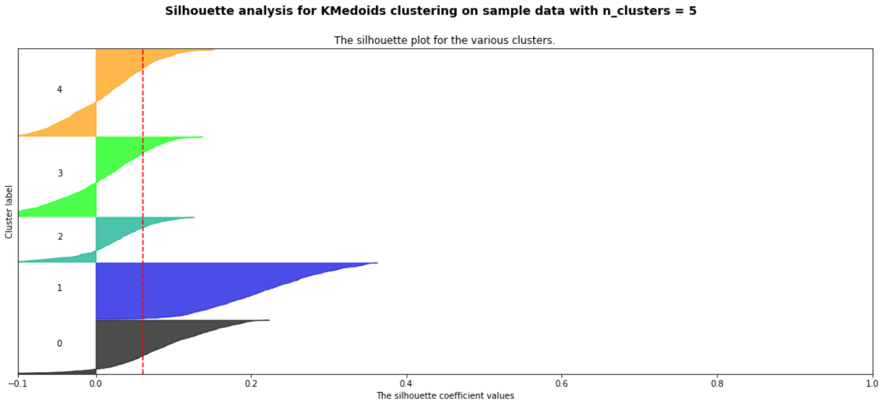


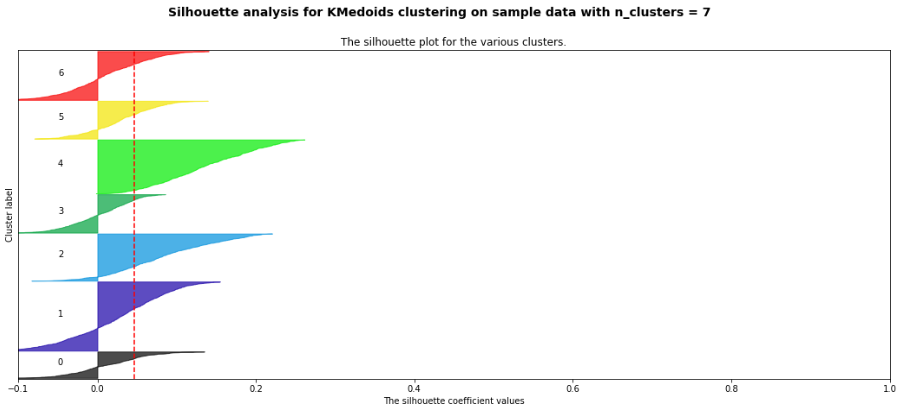


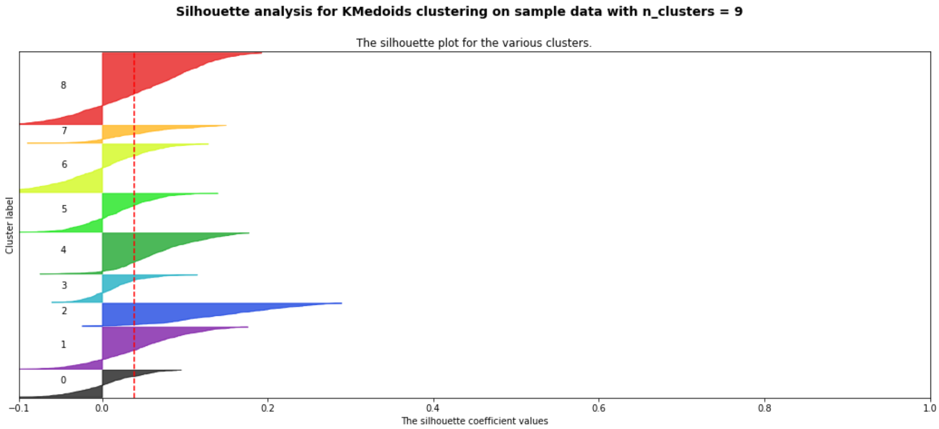
By looking at the elbow analysis, the most relevant K values resulted to be: 3,5,7,9,11. In this case the chosen k values are different from the ones of the architecture's literature.

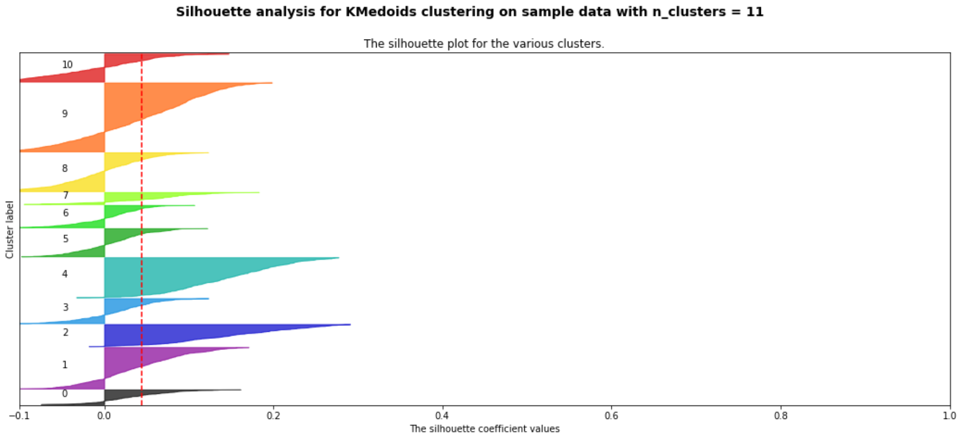
Silhouette Analysis







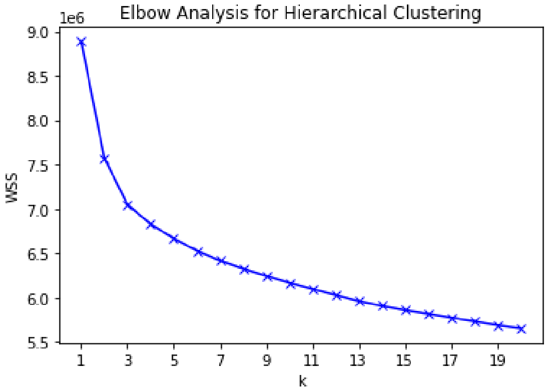


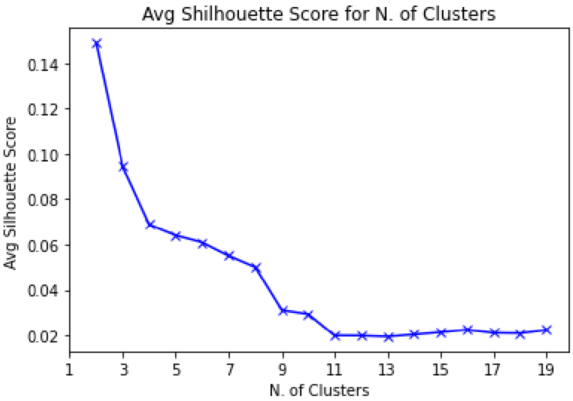


Also in this case, it is possible to conclude that the best clustering results are obtained for smaller values of K. The most important pattern that can be noticed is that with greater values of K, the percentage of points with negative silhouette score increases in each cluster.

Hierarchical Clustering

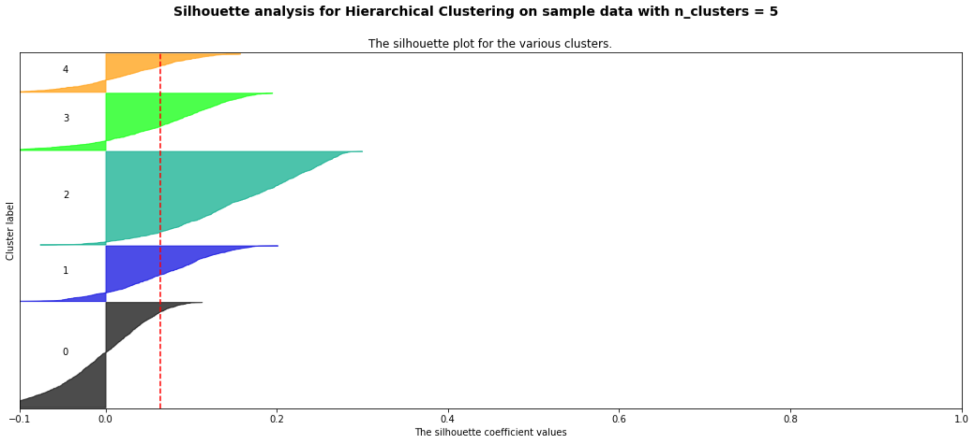
This algorithm creates clusters in a hierarchical way. It recursively merges pairs of clusters of sample data.

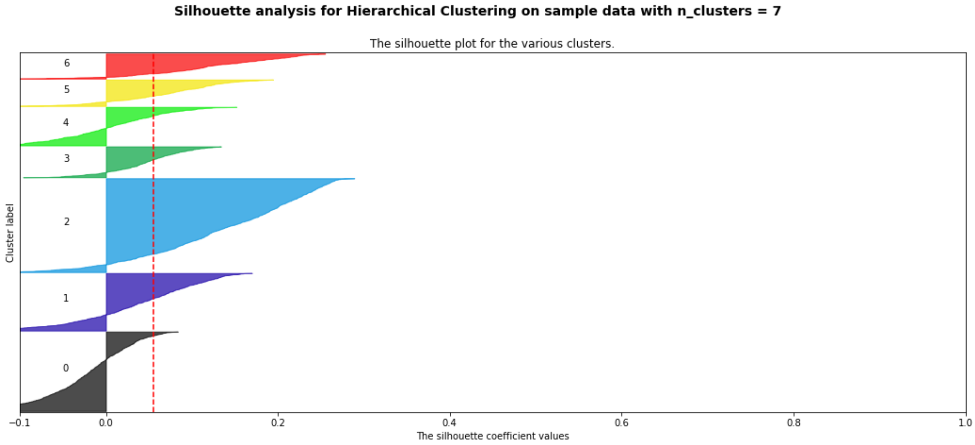


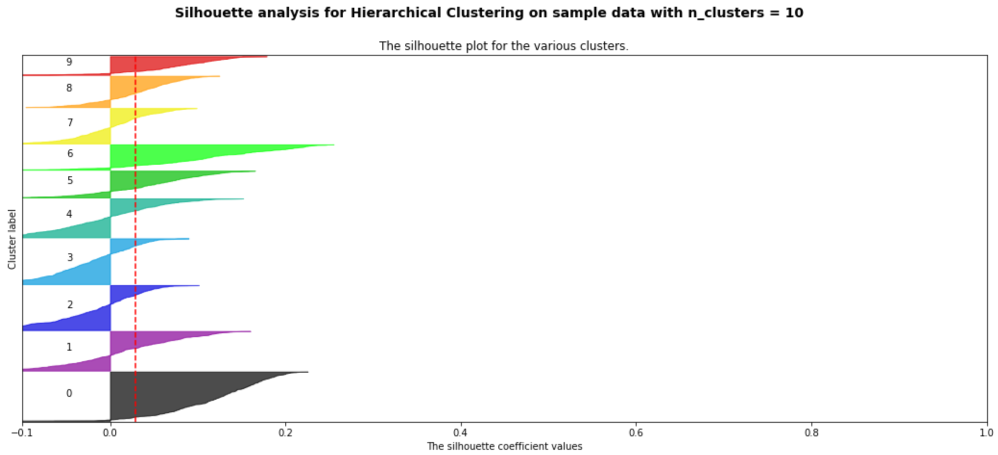


In hierarchical clustering, the elbow analysis has not provided significant insight regarding the most relevant K values, therefore the ones from the architecture literature have been chosen (k = [5, 7, 10]).

Silhouette Analysis





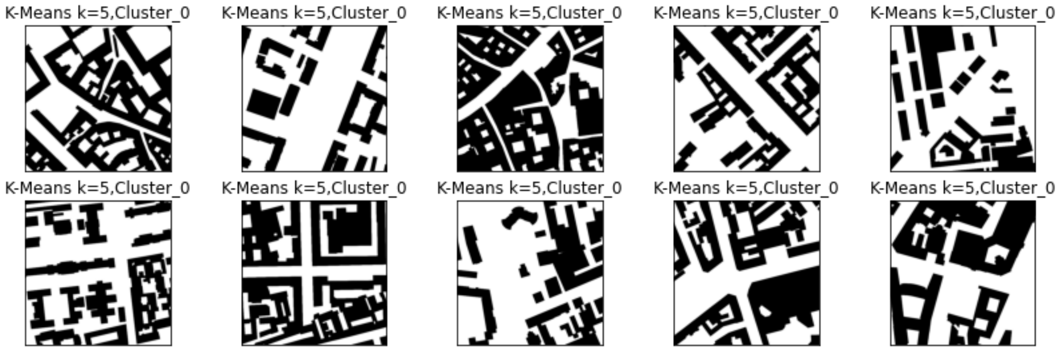


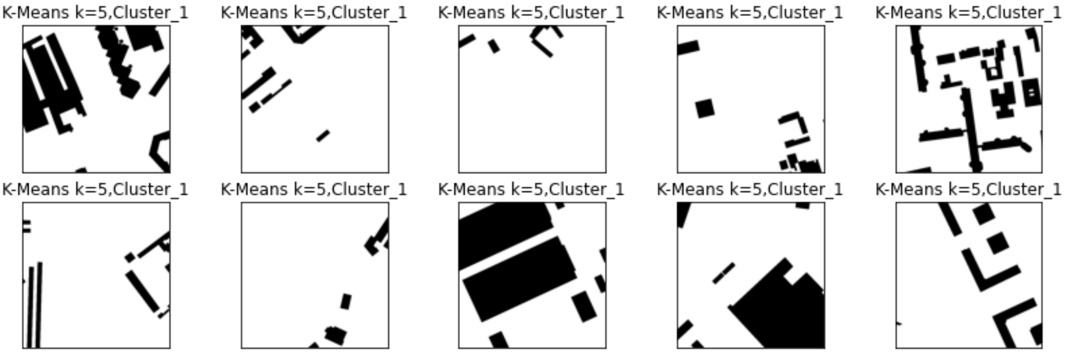
As in the previous cases, using smaller k values result in a higher silhouette score. Anyway, the average resulted to be relatively low in all the values of K.

Graphical Results

Here are shown some examples as a result of the clustering algorithms. Sections are divided by algorithm and each algorithm has been printed out with a specific value of k. Each plot shows some of the images belonging to a cluster. By opening the images folder it is possible to see all the images of all the clusters. Some differences and similarities are analyzed among the proposed images. Doing this analysis is important to understand why images belong to the same cluster (similarities) and why they belong to different clusters (dissimilarities).

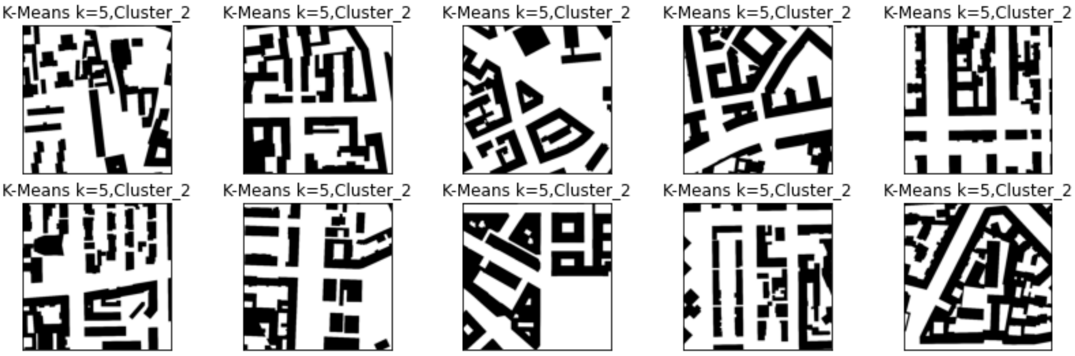
K-Means with K=5





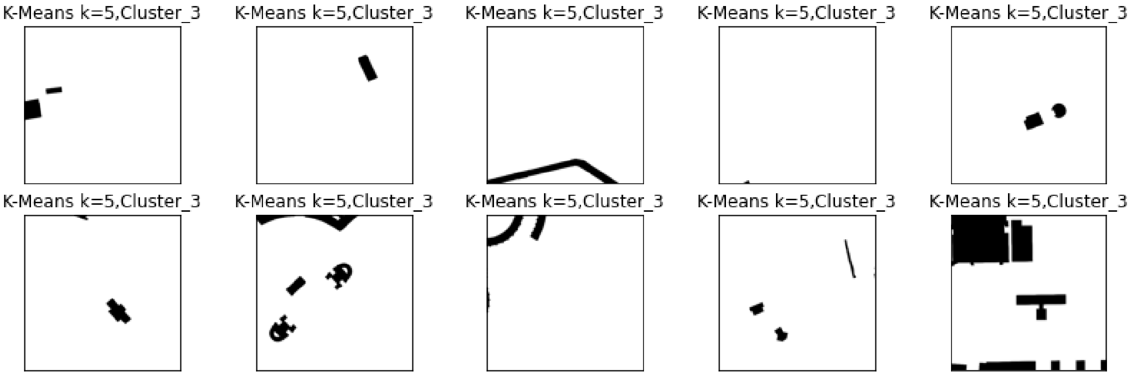
K-Means k=5,Cluster_1 K-Means k=5,Cluster_1 K-Means k=5,Cluster_1 K-Means k=5,Cluster_1 K-Means k=5,Cluster_1

K-Means k=5,Cluster_1 K-Means k=5,Cluster_1 K-Means k=5,Cluster_1 K-Means k=5,Cluster_1 K-Means k=5,Cluster_1



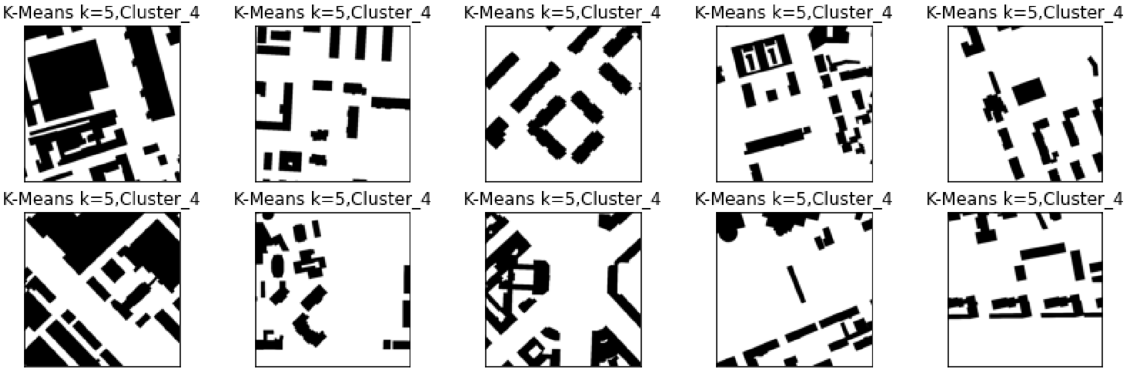
K-Means k=5,Cluster_2 K-Means k=5,Cluster_2 K-Means k=5,Cluster_2 K-Means k=5,Cluster_2 K-Means k=5,Cluster_2

K-Means k=5,Cluster_2 K-Means k=5,Cluster_2 K-Means k=5,Cluster_2 K-Means k=5,Cluster_2 K-Means k=5,Cluster_2



K-Means k=5,Cluster_3 K-Means k=5,Cluster_3 K-Means k=5,Cluster_3 K-Means k=5,Cluster_3 K-Means k=5,Cluster_3

K-Means k=5,Cluster_3 K-Means k=5,Cluster_3 K-Means k=5,Cluster_3 K-Means k=5,Cluster_3 K-Means k=5,Cluster_3

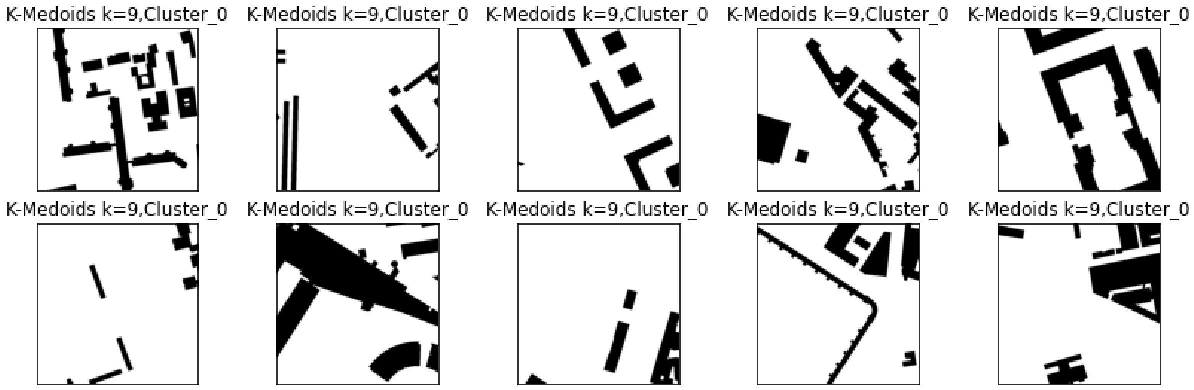


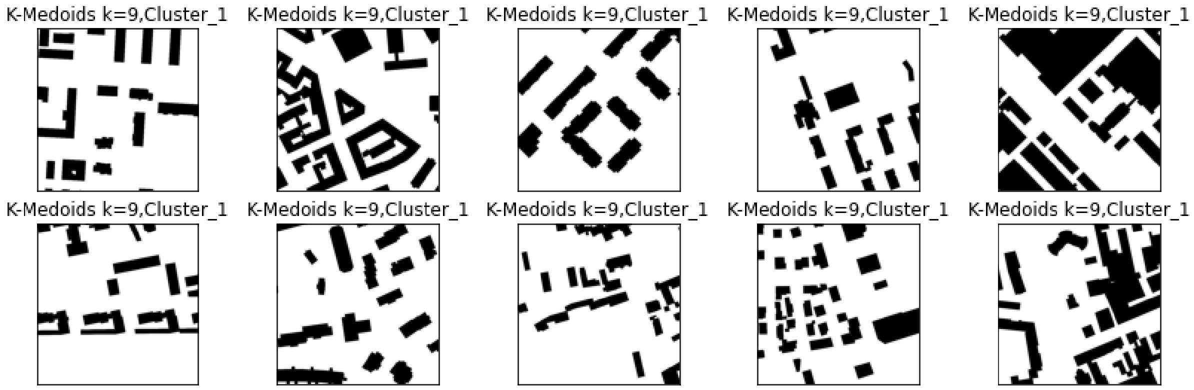
K-Means k=5,Cluster_4 K-Means k=5,Cluster_4 K-Means k=5,Cluster_4 K-Means k=5,Cluster_4 K-Means k=5,Cluster_4

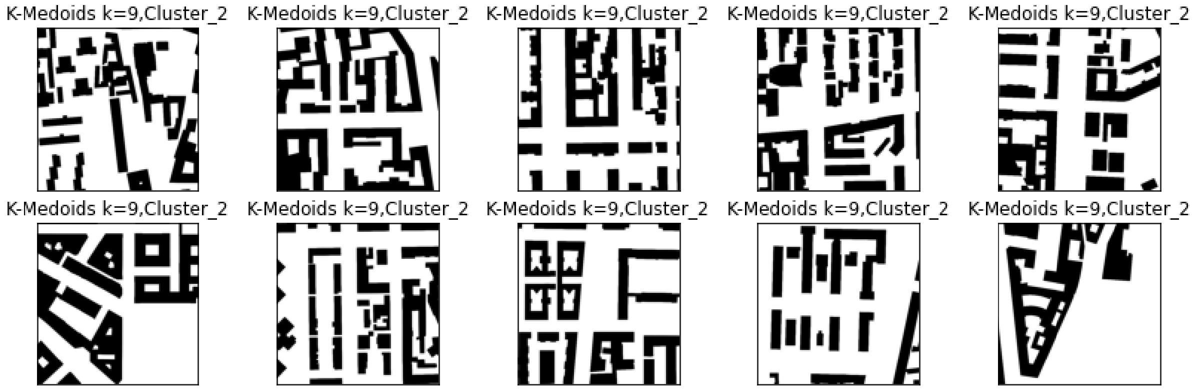
K-Means k=5,Cluster_4 K-Means k=5,Cluster_4 K-Means k=5,Cluster_4 K-Means k=5,Cluster_4 K-Means k=5,Cluster_4

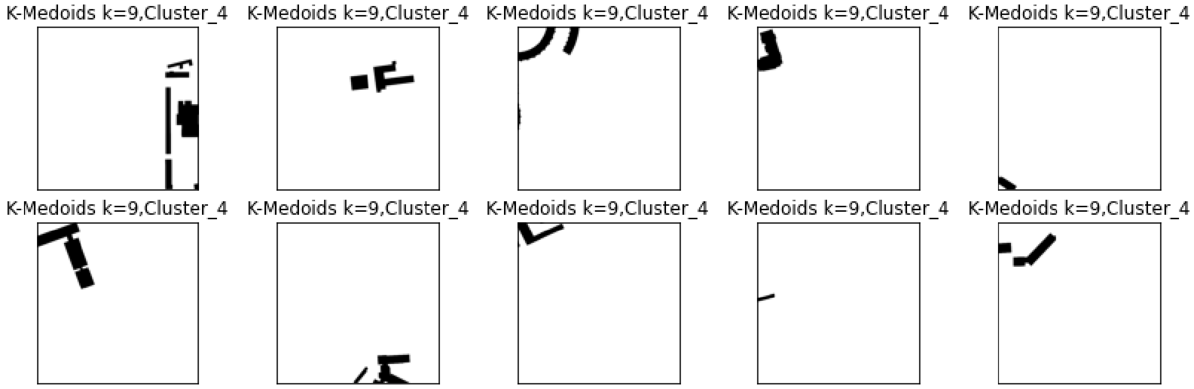
It is possible to notice that some clusters, like C0 and C2 show areas of the city with a high density of buildings. By looking at the images, it is possible to see that C2 shows buildings with an internal yard, while this aspect in C0 is less evident. Another difference among these two clusters is that C0 has a more irregular structure compared to the one of C2. Another pattern that may be noticed in C2 is that buildings are organized in blocks. In C3 there are very sparse buildings, maybe related to small structures in parks or green areas.

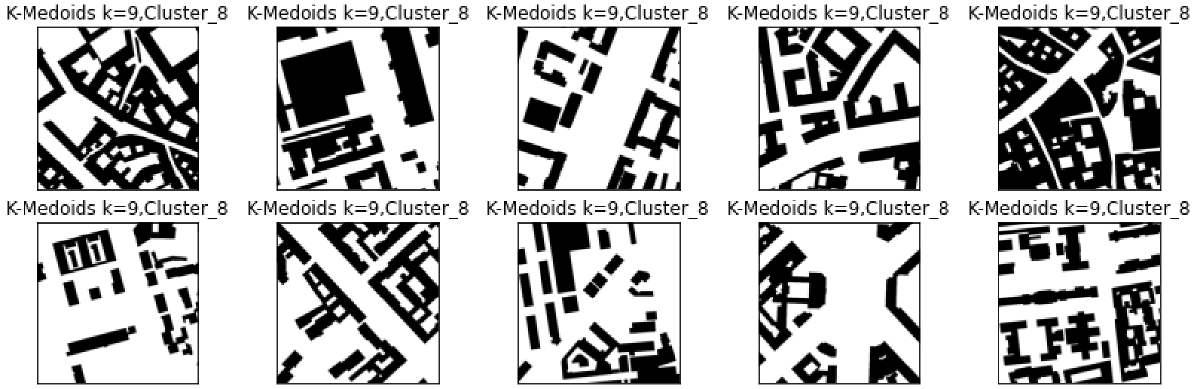
K-Medoids with K = 9







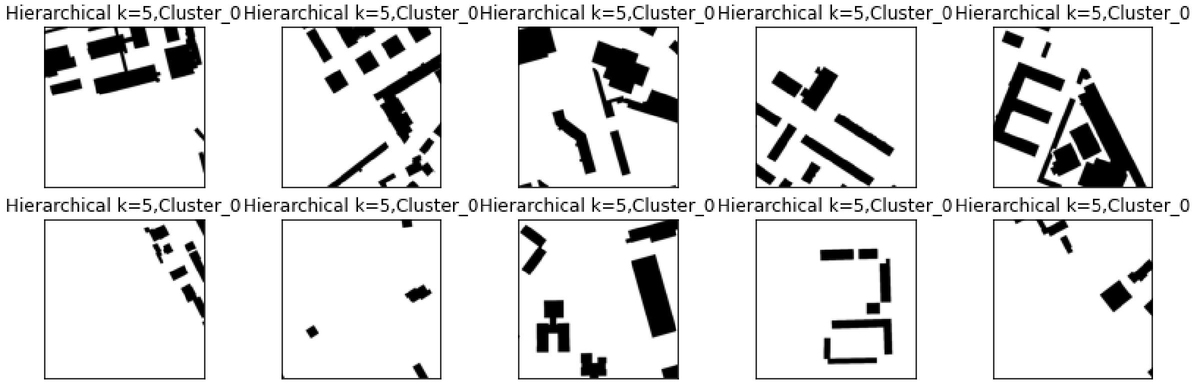


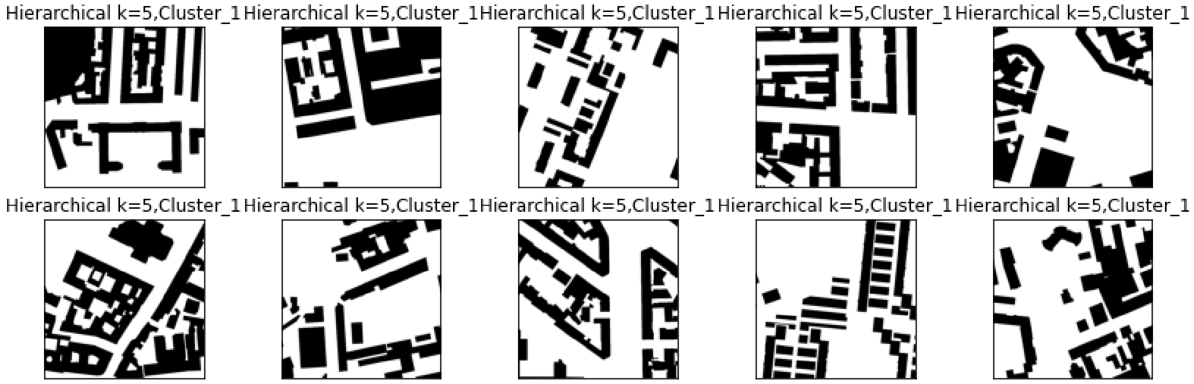


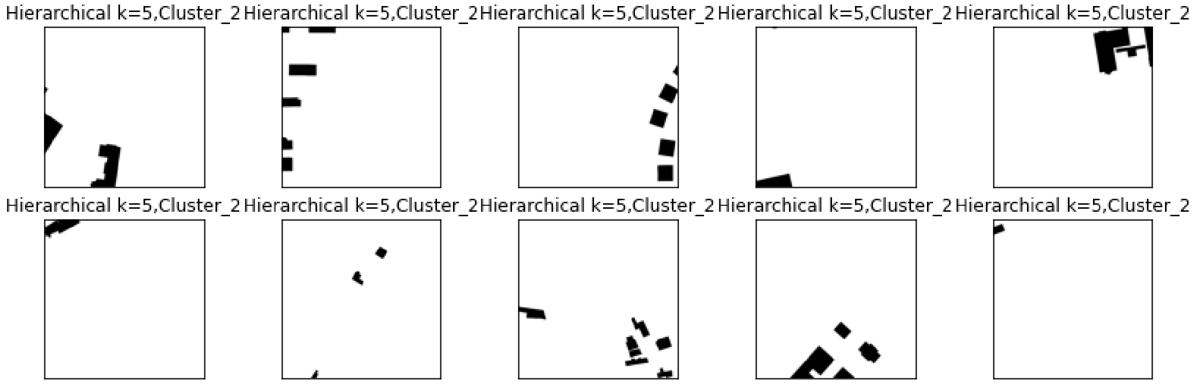
In this case, it is possible to see that C0 and C1 may represent areas with buildings separated from each other, maybe referred to suburban areas, where there are many separated tall buildings. By looking at the pictures, it seems that all the buildings do not have an internal yard. C4 could represent areas with very sparse buildings of random shape.

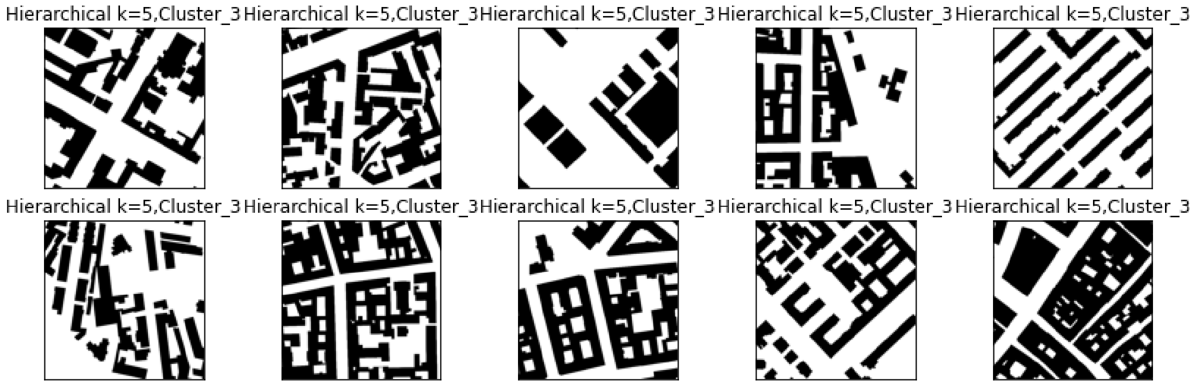
Also, C8 may represent parts of the grid with big roads in high building density areas, it is possible to see a crossroad in the example images. Also C2 seems to represent high building density areas, but buildings in C2 seem to be more geometrically structured compared to the ones in C8.

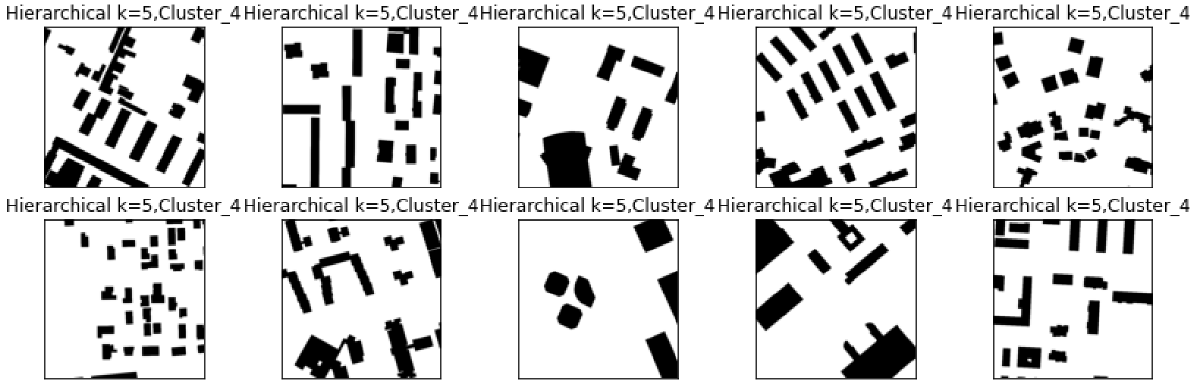
Hierarchical Clustering with K = 5


Hierarchical k=5,Cluster_0
Hierarchical k=5,Cluster_0
Hierarchical k=5,Cluster_0
Hierarchical k=5,Cluster_0
Hierarchical k=5,Cluster_0
Hierarchical k=5,Cluster_0
Hierarchical k=5,Cluster_0
Hierarchical k=5,Cluster_0
Hierarchical k=5,Cluster_0
Hierarchical k=5,Cluster_0


Hierarchical k=5,Cluster_1
Hierarchical k=5,Cluster_1
Hierarchical k=5,Cluster_1
Hierarchical k=5,Cluster_1
Hierarchical k=5,Cluster_1
Hierarchical k=5,Cluster_1
Hierarchical k=5,Cluster_1
Hierarchical k=5,Cluster_1
Hierarchical k=5,Cluster_1
Hierarchical k=5,Cluster_1


Hierarchical k=5,Cluster_2
Hierarchical k=5,Cluster_2
Hierarchical k=5,Cluster_2
Hierarchical k=5,Cluster_2
Hierarchical k=5,Cluster_2
Hierarchical k=5,Cluster_2
Hierarchical k=5,Cluster_2
Hierarchical k=5,Cluster_2
Hierarchical k=5,Cluster_2
Hierarchical k=5,Cluster_2


Hierarchical k=5,Cluster_3
Hierarchical k=5,Cluster_3
Hierarchical k=5,Cluster_3
Hierarchical k=5,Cluster_3
Hierarchical k=5,Cluster_3
Hierarchical k=5,Cluster_3
Hierarchical k=5,Cluster_3
Hierarchical k=5,Cluster_3
Hierarchical k=5,Cluster_3
Hierarchical k=5,Cluster_3


Hierarchical k=5,Cluster_4
Hierarchical k=5,Cluster_4
Hierarchical k=5,Cluster_4
Hierarchical k=5,Cluster_4
Hierarchical k=5,Cluster_4
Hierarchical k=5,Cluster_4
Hierarchical k=5,Cluster_4
Hierarchical k=5,Cluster_4
Hierarchical k=5,Cluster_4
Hierarchical k=5,Cluster_4

Also in this case the clusters seem to be created based on the same kind of patterns of the previous algorithms. This makes sense because even if the clustering algorithm is different, the extracted features are the same. For example both C0 and C4 may refer to areas with sparse buildings without internal yards. The difference between C0 and C4 seems to be that C4 contains many small buildings while C0 contains few but connected buildings. The cluster C3 seems to have many buildings with internal areas and also represents roads. Another thing that can be noticed is the fact that in C3 the buildings are organized in blocks. Finally, C2 still represents areas with low building density, but still a way lower compared to C0 and C4.

Discussion

In all the clustering algorithms it seems that the best patterns that are caught in the clustering regard the density of buildings, the presence of internal courtyards and the presence of roads. Regarding the elbow analysis, it has been noticed that in K-means and Hierarchical Clustering algorithms the curve was quite smooth and it prevented the choice of an appropriate k. In these cases the values of K were picked from the architecture's literature. K-medoids seemed to be the only algorithm that performed better, also from the analysis' perspective. In fact, it is the only algorithm in which the values of K were not chosen from the literature. Regarding the silhouette analysis, it has been noticed that in all the three cases the score was decreasing when K was increasing. In fact, when the K was increasing there were many more points with a negative score. Usually, having an average silhouette score close to zero means that clusters overlap with each other. It could be our case especially when K increases.
This could make sense because the VGG16 extracts general patterns from the images and it leads to cluster images for few but evident similarities. In fact, by looking at the graphical results the most common patterns among images were related to building density, the presence of internal yards and the presence of roads. The main patterns that the network extracted seem to be related to: the amount of black and white in the images, the structure of the buildings, their shape, how they are organized with each other. Moreover, the extraction of high-level patterns agrees with the fact that the highest score for the elbow and silhouette analysis was obtained for lower values of K. On the other hand, having patterns that are too precise and peculiar of a data point would lead each point to be considered as a cluster itself.

Future Work

The most important and crucial points are the feature extraction part and the choice of the clustering parameters. In order to improve the feature extraction part, the last fully connected layers could be trained over the dataset in order to have a CNN that captures general patterns in the first layers and more task oriented patterns in the last ones. This was not done due to the small size of the dataset compared to the size of the network.
Another strategy would be to train the network by minimizing the loss function based on the clustering feedback. This would allow a better feature extraction for the specific task, that is clustering. Further improvement could consider the initialization of the centroids of the clusters and other metrics used in the assignment of points to clusters.
Furthermore, trying other clustering algorithms might be another option to increase the performance. In fact, both K-means and K-medoid algorithms create circular shape clusters and this may represent a limit to express the actual way in which images group together. Other density based approaches like DBSCAN or HDBSCAN might be experimented.
Finally, a greater number of samples would have helped in the neural network training, in order to let the network extract more specific and task oriented patterns. In fact, this was the most important problem that had to be tackled. One solution that could be used to increase the size of the dataset would be the usage of data augmentation, that applies graphical changes to images in order to generate new ones. In conclusion, the project allowed an exploration of the patterns extracted by a neural network in this particular field of architecture and provided a comparison of the clustering results.