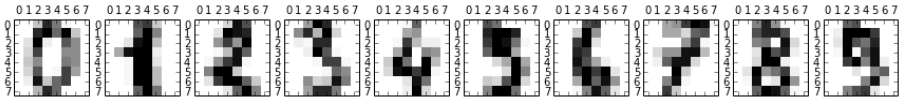
Practical Task 16

In this task you will work with images. It's convenient to use SciKit-Image package. Install PIL to read JPG images. If your computer is running extremely slow on original images, you're allowed to downscale the image to at least 200 x 200 pixels.

Recognize handwritten digits in an unsupervised manner



1. Load the dataset $digits$ using the function $sklearn.datasets.load\_digits$.
2. Cluster the images with $KMeans$ algorithm. Run KMeans with different initializations. You cannot use labels information for initialization. Do not forget to shuffle images before running KMeans!

   Useful functions: $sklearn.cluster.KMeans$.
3. Using the labels, evaluate the quality of different clusterizations using $Adjusted\ Mutual\ Information$ and $Adjusted\ Rand\ Index$. Take into account that one of the algorithm's parameters is initialization type ($init$) – you should also compare the clusterization quality using different values of this parameter. Is there a correlation between quality metrics values?

   Useful functions: $sklearn.metrics.adjusted\_mutual\_info\_score$, $sklearn.metrics.adjusted\_rand\_score$.
4. Visualize the images that correspond to the centroids of the best clusterization.

   Useful functions: $matplotlib.pyplot.imshow$.
5. Can you reach ideal separation of digits? Which images were assigned to a wrong cluster (we consider the image to be in a wrong cluster if its true label differs from the most popular label in that cluster)? Give the examples of such images and try to explain why that errors happened.

Limit the color pallet of the image with minimal loss in visual quality

1. Load the image $parrots.jpg$.

   Useful functions: $skimage.io.imread$ (if you have a read error, you should verify that you have Pillow package installed).

2. We propose to do a clustering of pixels in RGB space in order to limit the color pallet of the image. After clusterization all pixels assigned to one cluster are filled with one color, that color could be a centroid of the corresponding cluster or a median color in the cluster.

3. Using a KMeans algorithm cluster the image colors in RGB space. Visualize clusterizations with number of clusters K = 2, 3, 10, 20, 25, 30.

4. Starting with what K value the clusterization result looks acceptable to you?

Segment the image in semantic components

1. Load the image $grass.jpg$.

2. Find segments on the image using clusterization of pixels in the space $\psi_i = [\lambda x_i, \lambda y_i, r_i, g_i, b_i]$, where (x_i, y_i) – pixel coordinates, (r_i, g_i, b_i) – pixel color, λ – parameter denoting the importance of spacial connectivity compared to color similarity.

3. How the λ parameter and the number of segments influence the segmentation result?

4. Visualize the segmentation results (as on the image below) for different values of parameters.

   Useful functions: $matplotlib.pyplot.contour$ (the usage example can be found in SciKit-Learn "Segmenting the picture of Lena in regions").

   You are not asked to measure the segmentation quality in this subtask, only images and conclusions are expected in the result.

5. Try to find segments using spectral clustering.

   Useful functions: $sklearn.cluster.spectral\_clustering$.

6. Experiment with algorithm's parameters. How the $assign\_labels$ parameter and the number of segments influence the segmentation result?

7. Compare to algorithms and their resulting segmentations.