lab02-recognition-assignment

Joanna Erika Brodbeck, Computer Science, Bsc 7. Semester
jbrodbec@student.ethz.ch

Abstract

Here is the report for the lab02-recognition-assignment.

For this exercise, Copilot was used to help with the syntax. The written report was checked with Grammarly.

1 K-means Clustering

1.1 Initialisation

Values for $k \in \{2, 3, 5, 6, 8, 10\}$ have been tested. Depending on what kind of results is needed, every k has its strengths and weaknesses. For the image of ETH, $k = 3$ is enough to segment the most important parts, sky, building and street. $k = 10$ does not give any clear information in this scenario. It is a question of balance how to choose k but it should be in mind not to chose it as too big and not too small.

1.2 Point Distance Calculation

The helper function for the Euclidean distance has been implemented.

$$d(\mathbf{a}, \mathbf{b}) = \sqrt{\sum_{i=1}^{n} (a_i - b_i)^2}$$

In Python, this can be implemented as follows:

```
def euclidean_distance(a, b):
    return np.sqrt(np.sum((a - b) ** 2))
```

Here, $\mathbf{a}$ and $\mathbf{b}$ are numpy arrays representing points in n-dimensional space.

1.3 K-means Iteration

The loop in kmeans runs either a maximum of iterations or until convergence is reached. The function has several steps:

- Step 1: Assign pixels to centroids:
  - For each pixel_values, the distance to each centroid is calculated. This is done with the previously implemented function euclidean_distance.
  - After iterating, the pixel is assigned to the cluster corresponding to the nearest centroid.
- Step 2: Updating Centroids:
  - The After assigning all the pixels to the corresponding centroids, a new mean and therefore a new cluster is calculated.
  - This is done by gathering all points that belong to a particular cluster
  - Then the new centroid for each cluster is calculated as the mean of the points assigned to it.
  - If no points are assigned to a cluster, then it is not updated and skipped.
- Step 3: Convergence:
  - If all the old centroids are the same as the new centroids, the loop breaks and returns the new labels and centroids.
  - If something changes, it goes to Step 1 again until the maximum iteration is reached.

1.4 Experiment with Different Values of K

For the k-mean clustering, several k's have been tried out to see the differences. The result can be seen here Figure 1.
In the first image, one can see two main clusters, dark and bright. As the clusters increase one can see more details in shades. Already with five clusters, the sky for example is extracted into blue and white, indicating different shades of colour.

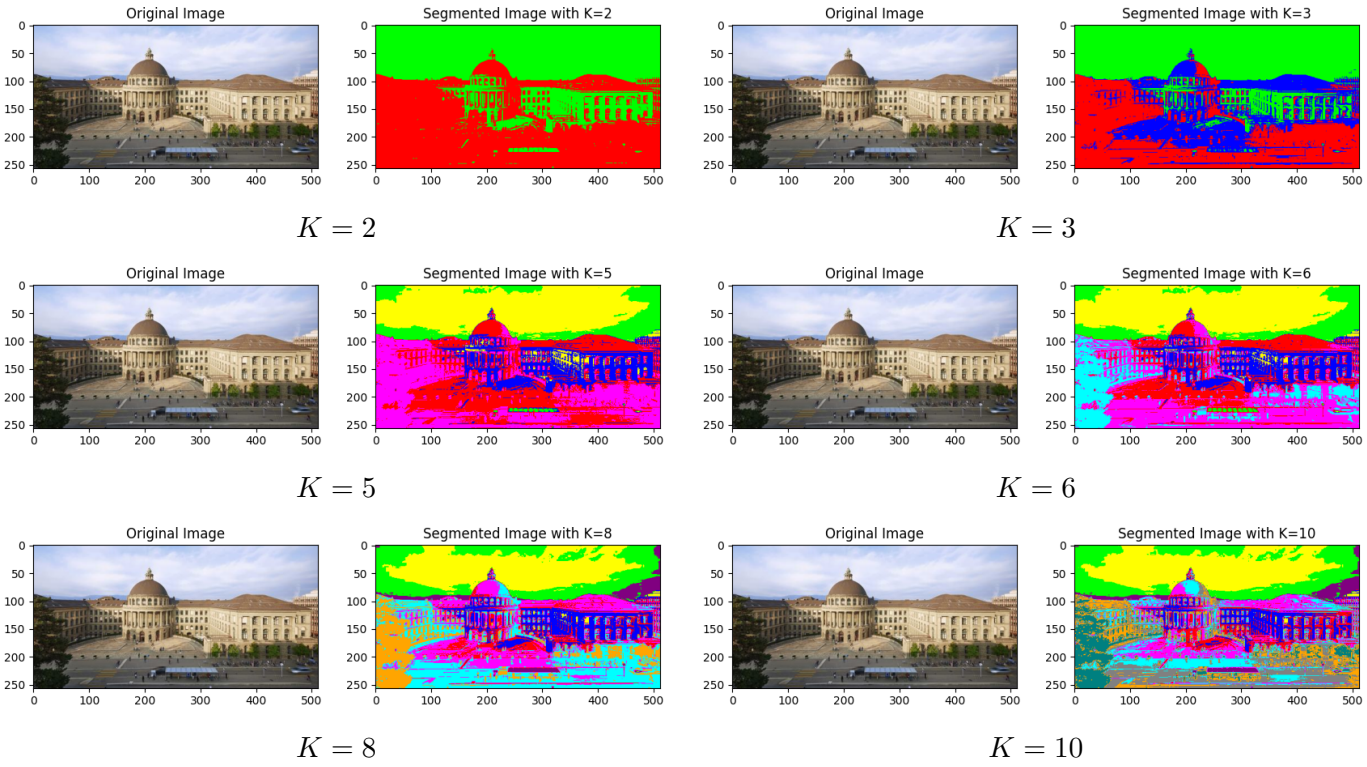

Figure 1: The image with different k-mean clustering

The computational effort cost increases significantly. This is because several loops are used where it iterates through all the $k \in K$. So, for $K = 10$, the distance to each of the 10 possible centroids is calculated for every pixel. This leads to a significantly higher cost than if the loop only needs to run twice for $K = 2$. However, the result is less detailed than the ones with many clusters. It is important to have a balance of computational efficiency and accurate results. It does not gain any advantages if K is too big, as the original image can then be used. If K is too small, no real information is gathered and no insightful conclusion can be made.

2 Bag-of-words Classifier

2.1 Local Feature Extraction

2.1.1 Feature detection - feature points on a grid

The function calculates how big the new image should be, this is in x and y direction $2 * \text{border}$. Then the cell size is calculated with the new image size divided by the number of points it should contain.
Then the $vPoints$ are filled with the coordinates, by iterating through the nPointsX and nPointsY and calculating the correct size.

2.1.2 Feature description - histogram of oriented gradients

The implementation is structured into several steps: gradient_magnitude, gradiet_orientation and the histogram creation with the previous results. (I read another tutorial where they used the magnitude of the grad_x and grad_y, which led to better results than just adding +1 to it www.analyticsvidhya.com). But essentially, the orientation of each grad batch was calculated and ensured that it is in the range of $0, 2\pi$. Then all the angles and magnitudes are calculated for each batch and then in the histogram added that it corresponds to. In the end, it is normalised and appended to the descriptors.
But after understanding what the different angles and magnitudes are doing and what the bins are exactly, I used the library $np.histogram()$.

2.2 Bag-of-words Vector Encoding

2.2.1 Bag-of-Words-histogram

For this, the function that was already given (findnn) was used to get the indexes of the closest features. In the histogram, the corresponding indexes were increased.

2.2.2 Processing a directory with raining examples

Here all the implemented functions are used to return vBow.

2.2.3 Nearest Neighbor Classification

1 if it is true (car) or 0 false, (no car). This was done by comparing the distances of findnn and if the DistPos is smaller than the DistNeg, there is a car and 0 otherwise.

3 Test Accuracy

Table 1: Results of different k and numiter values with accuracies for positive and negative classes.

k	**numiter**	**acc_pos**	**acc_neg**
5	10	0.7755	0.86
5	50	0.8980	0.88
5	100	0.8776	0.94
10	10	0.9796	0.98
10	50	1.0000	1.00
10	100	0.9388	1.00
15	10	1.0000	0.94
15	50	1.0000	1.00
15	100	1.0000	0.96
20	10	1.0000	0.96
20	50	1.0000	0.98
20	100	1.0000	0.94
25	10	0.9388	0.92

Several k and numiter have been tested to see how the results behave. It can be seen that $k = 10$ and $numiter = 50$ achieved good results. It can be described like this: if k is too small, then big junks of the image are considered as a dataset for the vocabulary. The positive accuracy will be low, but the negative accuracy high. This is because only similar junks will be considered as cars, as the vocabulary is already very big. On the other hand, if k is too big, everything will look like a car as the vocabulary is too big. The positive accuracy will be high, as everything will be considered as a car, but the negative accuracy will be low. This can also be seen in the table Table 1