Computer Vision Report UE 1 - Group 12:

Tomas Musil (1167504)
Andreas Wittmann (1225854)
Hanna Huber (0925230)

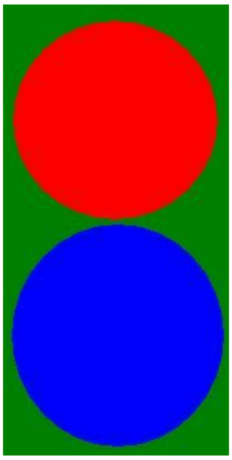
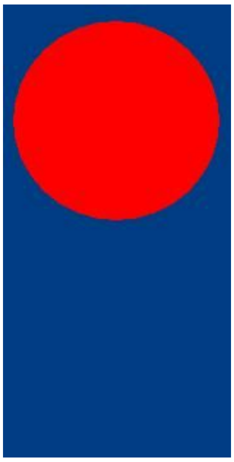
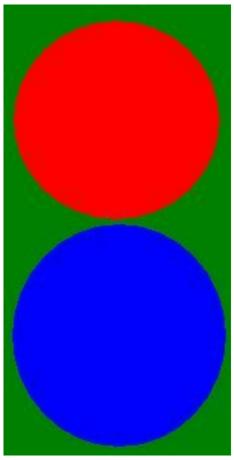
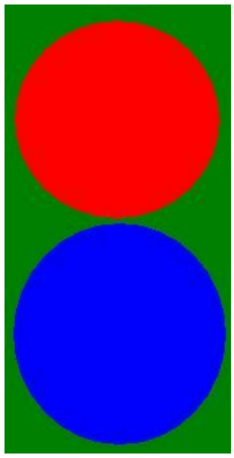
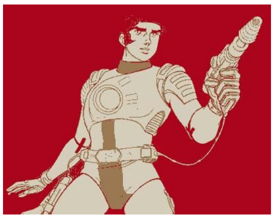
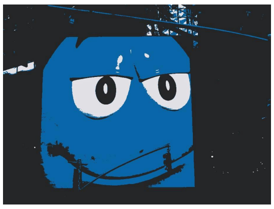
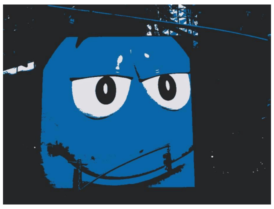
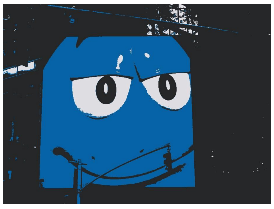
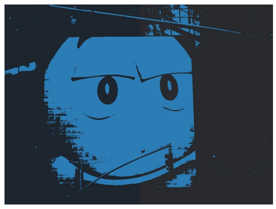
Task 2: K-Means

Show the results for all images in the case of 3D data points as well as 5D data points (using a fixed value of K). Discuss the results. Which data representation is better in your opinion?

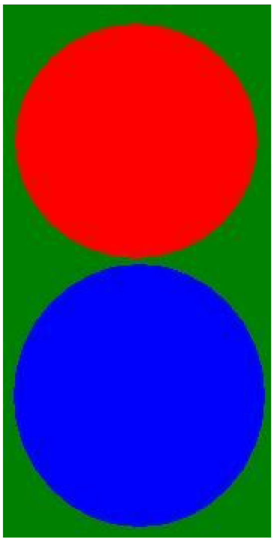
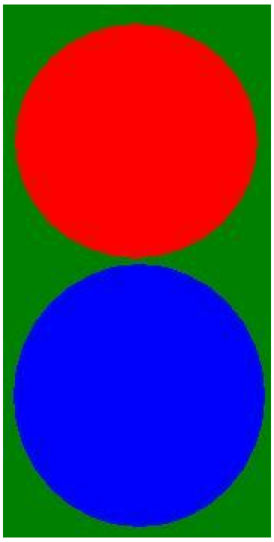
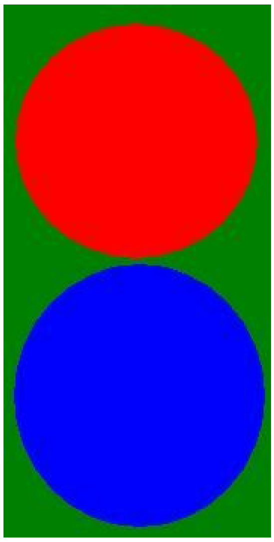
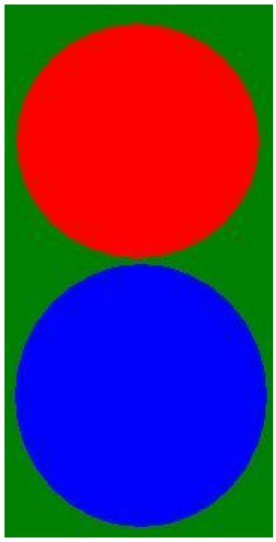
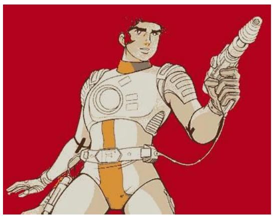
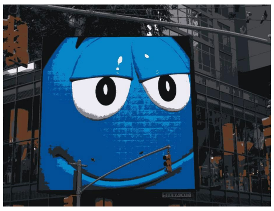
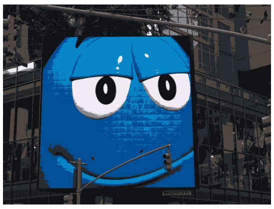
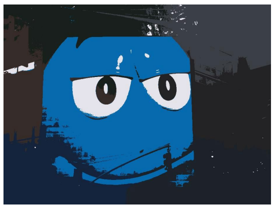
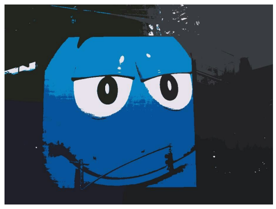
We have tested the k-means algorithm with all sample images using a k value of 3 and 8. The 3D k-means algorithm shows better results than the 5D k-means algorithm in both cases. At the beginning of our test, we thought that the k-means algorithm with spatial information would deliver way better results. During our analysis of the results, we notice that the spatial information of the 5D k-means algorithm is a drawback especially, if the k value increases. Moreover, the spatial information is a disadvantage, if the image contains a lot of low frequencies or if pixels of the same color value are distributed over the whole image. (e.g.: “future.jpg”, “mm.jpg”)
Nevertheless we want to mention that the 5D version of the k-means algorithm overrules the 3D version in one case. If the image contains coherent clusters like in the “simple.png” sample and the k value is exactly the value of existing clusters, the 3D algorithm could deliver wrong results, while the 5D algorithm delivers the right result at each execution. This could be explained by the randomly chosen starting points of the algorithm.

Examples

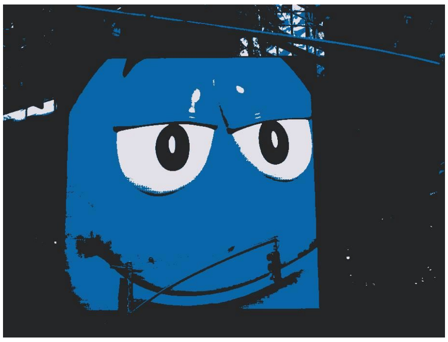
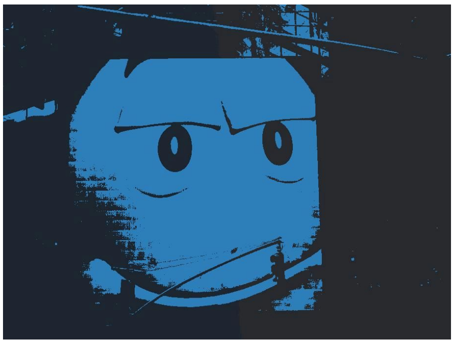
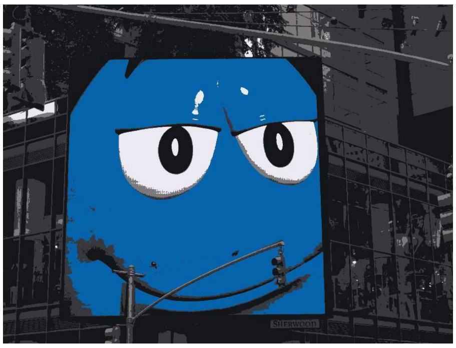
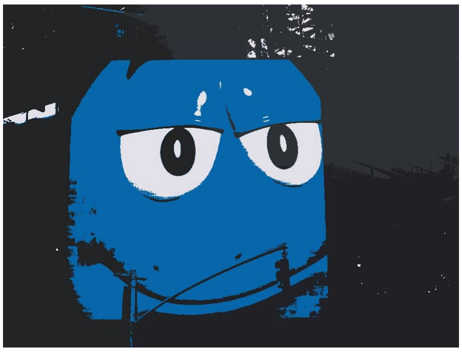
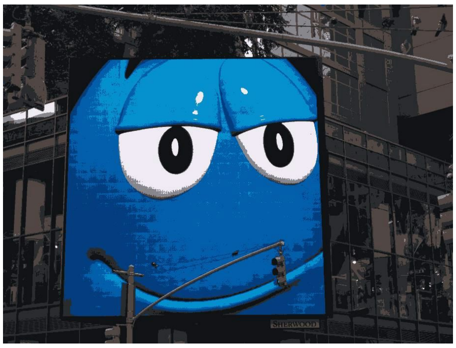
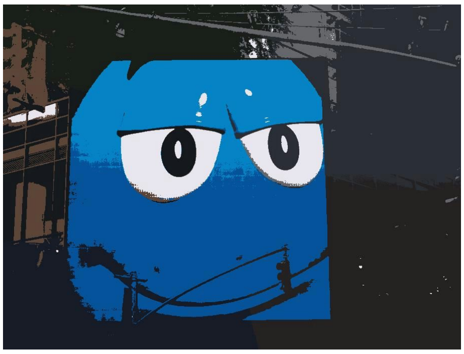
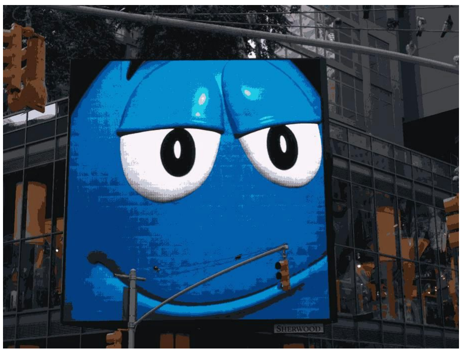
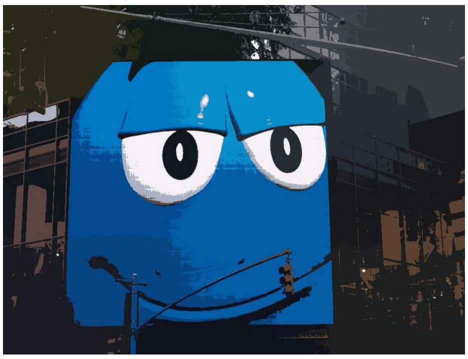
k=3:

3D - first run	3D - second run	5D - first run	5D - second run
			
			
			

k=8:

3D - first run	3D - second run	5D - first run	5D - second run
			
			
			

Apply different values of K to the image mm.jpg and show the results for both 3D and 5D data points. Interpret the results.

k	3D	5D
3		
5		
10		
20		

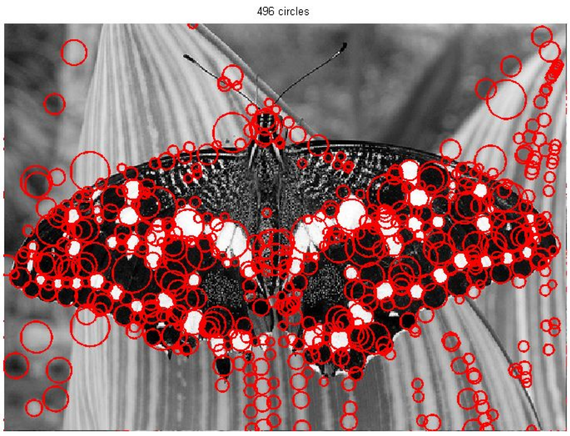
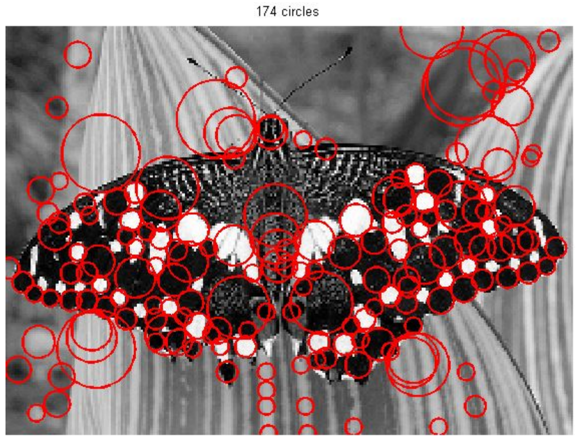
Again, using 5D datapoints leads to coherent clusters as expected. On the contrary, for this example image, 3D datapoints produce semantically correct results. For instance, the traffic light and the post holding it are already connected for k=5, whereas with 5D datapoints they are still disconnected for k=20.
Thus, semantic segmentation can generally be performed with fewer clusters if 3D datapoints are used. However, if two initial centroids happen to fall into a region that should be classified as a single cluster, it might be divided between two similar clusters (see table 1, k=3, 3D, second run).

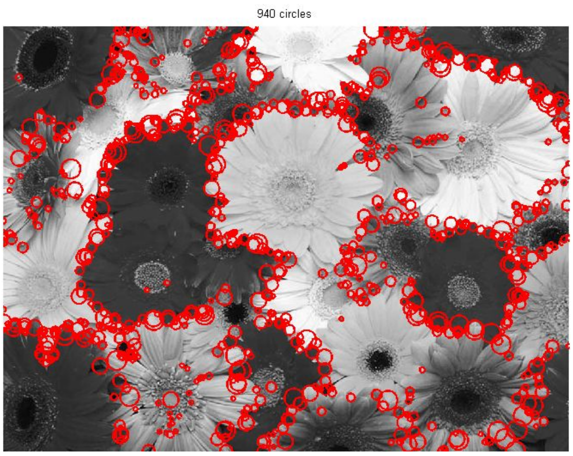
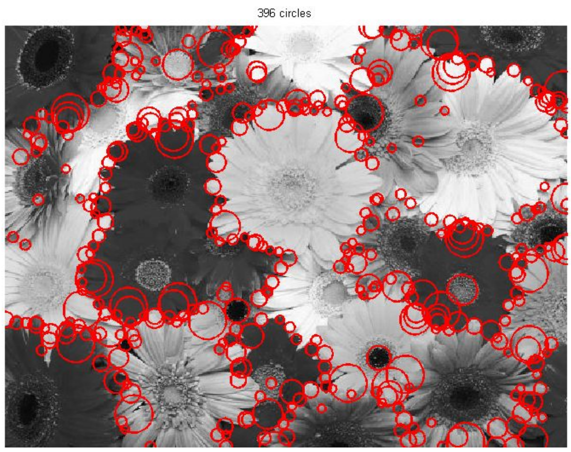
Where do you see - based on your results - the strengths and the weaknesses of the method?

Table 1 indicates that the method strongly depends on the initial values for the cluster centroids which are chosen randomly. This represents a possible weakness. However, if suitable initial values are used, plausible results can already be achieved with few clusters if 3D datapoints are used. 5D datapoints should only be used if prior information about the image or the clusters you want to detect is available that indicates that clusters should be coherent.
Moreover, this method allows different clusters to have the same mean value, which might not always be desirable.

Task 3: Scale-Invariant Blob Detection

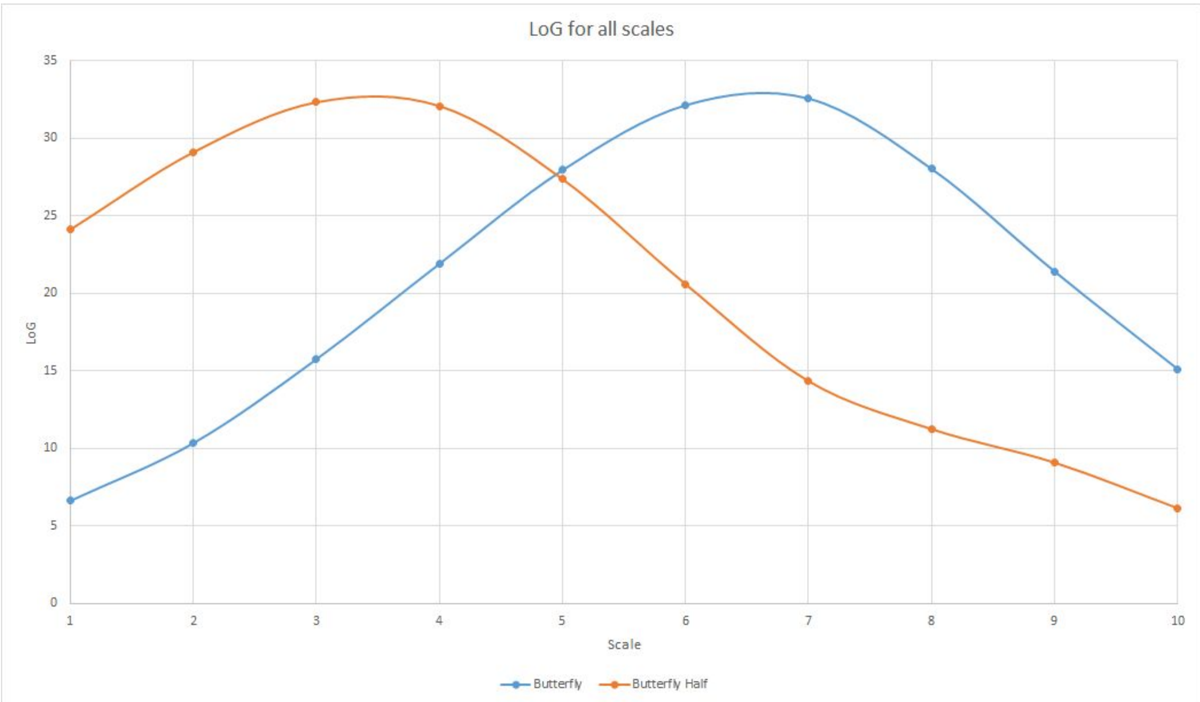
Apply the method to both the original images as well as to half-sized versions of them. Draw the detected blobs as circles with appropriate scale. Is the method able to find blobs in a scale-invariant way? If there are errors, what are the reasons for them?

493x356	247x178
	

1024x768	512x384
	

Yes, this method is able to find blobs in a scale-invariant way, because there were different sizes of blobs detected. For a constant threshold, the number of detected blobs decreases with the image size. The scale at which a corresponding blob is detected varies corresponding to the downsampling factor (see figure below).
The blob detection is restricted to scales that were used to create the scale space. This can be adjusted by modifying the inital sigma value and the number of scale space levels.
Results also depend on the parameters used for filter creation, e.g. boundary options. Here the parameter ‘replicate’ is used in order to avoid unsuitable values at image boundaries.

Pick a detected keypoint and plot the response of the LoG for all scales in both image versions. The outcome should be a 2D plot where the x-axis represents the scale of the filter and the y-axis the filter response at the selected keypoint position. Describe and explain the difference between the two curves.



In diagram there are two curves visualized, which correspond to a Gaussian distribution. Both of them have the same distribution, but shifted. The detected key point is positioned at the maximum of the curve, which corresponds to the scales 3 (maximum at 3.4) and 7 (maximum at 6.8), respectively. This factor corresponds to the downsampling factor.