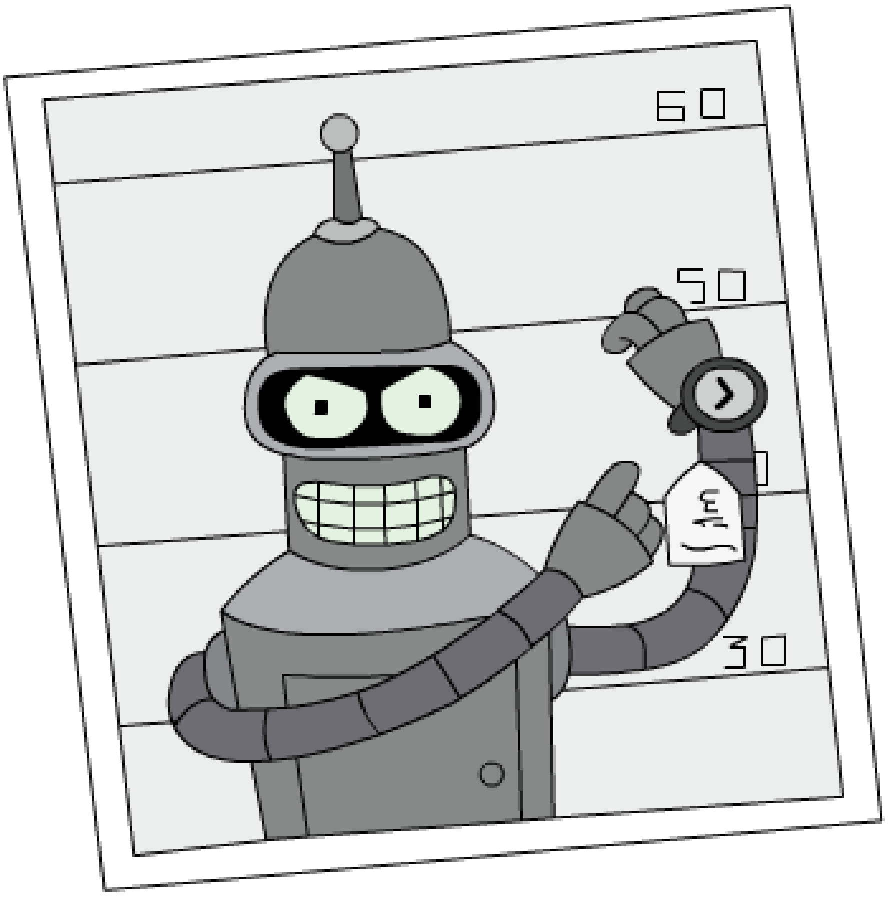
Kmeans, superpixels and region proposal

ECE4076 Computer Vision
Lab 3 (Weeks 8,9)



Academic integrity

Every lab submission will be screened for any collusion and/or plagiarism. Breaches of academic integrity will be investigated thoroughly and may result in a zero for the assessment along with interviews with the plagiarism officers at Monash University.

Late submissions

The default late submission university penalty will apply.

Lab Instructions and the Use of Generative AI

You may use numpy and matplotlib and opencv for image loading.

- You must implement kmeans clustering from first principles and should not rely on any pre-existing kmeans clustering libraries.
- You should use Matplotlib to display images and any intermediate results.
- You may use generative AI.

Lab Grading

Each lab is worth 8%, and there are a number of sections and tasks with their own weighting. A task is only considered complete if you can demonstrate a working program and show an understanding of the underlying concepts. Note that later tasks should reuse code from earlier tasks.
Marks will be provided based on the correctness of the code, the quality of your results, comments indicating you understand your work, and a discussion of tasks provided at the end of the notebook.

Kmeans clustering, Superpixels and Region Proposal

In this lab we are going to explore image clustering and superpixels segmentation. You will be required to complete the following tasks:

1. Implement k-means clustering to assign each pixel of an image to one of K distinct clusters, using colour similarity to determine assignment.
2. Modify your k-means clustering to find superpixels by assigning each pixel of the image to one of K distinct clusters, using both colour and location similarity to determine assignment.
3. Use superpixels to propose regions of interest in the image (we will learn about neural networks that use this information in week 11).
4. Write a brief report (max 600 words) discussing explaining your implementation and results. Explain your reasoning by referencing the intermediate result figures you generated.

References

- k-means clustering
- SLIC Suerpixels
- Week 7 Lecture content

Resources

- Lab3_student_template.ipynb - please complete the lab tasks in this template notebook and use the markdown spaces provided to report your findings/ describe your approach.
- CSIRO_ScienceImage_3831_Ulysses_Butterfly.jpg is located in the lab3 folder
- Example outputs are included in the student pack.

Figure 1: Sample image

Task 1: Implement K-means from first principles: colour similarity

Fill in the function provided to implement the following steps. Your function should accept an image, a variable storing the number of iterations to repeat k-means for, the number of clusters K.

1. Flatten the image into an Nx3 vector of pixels (num pixels x r,g,b colour of pixel)
2. Select K random colour centroids
3. Repeat for n iterations:
   - Assign each pixel in the vector to a cluster and store the index corresponding to this in a numpy array
   - Update the centroid locations using the new cluster assignments
   - Save the total loss at each iteration
4. Convert the cluster index array back into a 2D array the same size as the image, where each pixel contains the corresponding cluster index
5. Create a new quantised image, where each pixel is coloured according to the centroid corresponding to a given cluster index.
6. Return the loss, cluster index image and quantised image
7. Test your function on the sample image provided, and select an appropriate number of iterations
8. Display the results for a number of choices of K. Use the returned losses to select an appropriate value for K

Task 2: Modify K-means to cluster based on colour similarity and pixel location and extract superpixels

1. Modify the kmeans function above to segment based on colour and pixel location, so that you extract superpixels.
2. Test your implementation by clustering the image into 50 superpixels.
3. Select an appropriate pixel coordinate scaling factor so that you trade-off incorporating colour information into the clustering, while ensuring superpixel continuity (all pixels in a superpixel must remain connected).

Task 3: Region proposal

Segment your image into 10 superpixels. Then find the locations of the top left pixel and bottom right pixel in a superpixel, and use this information and the cv2.rectangle function to draw rectangles (regions) on the original image for each superpixel.

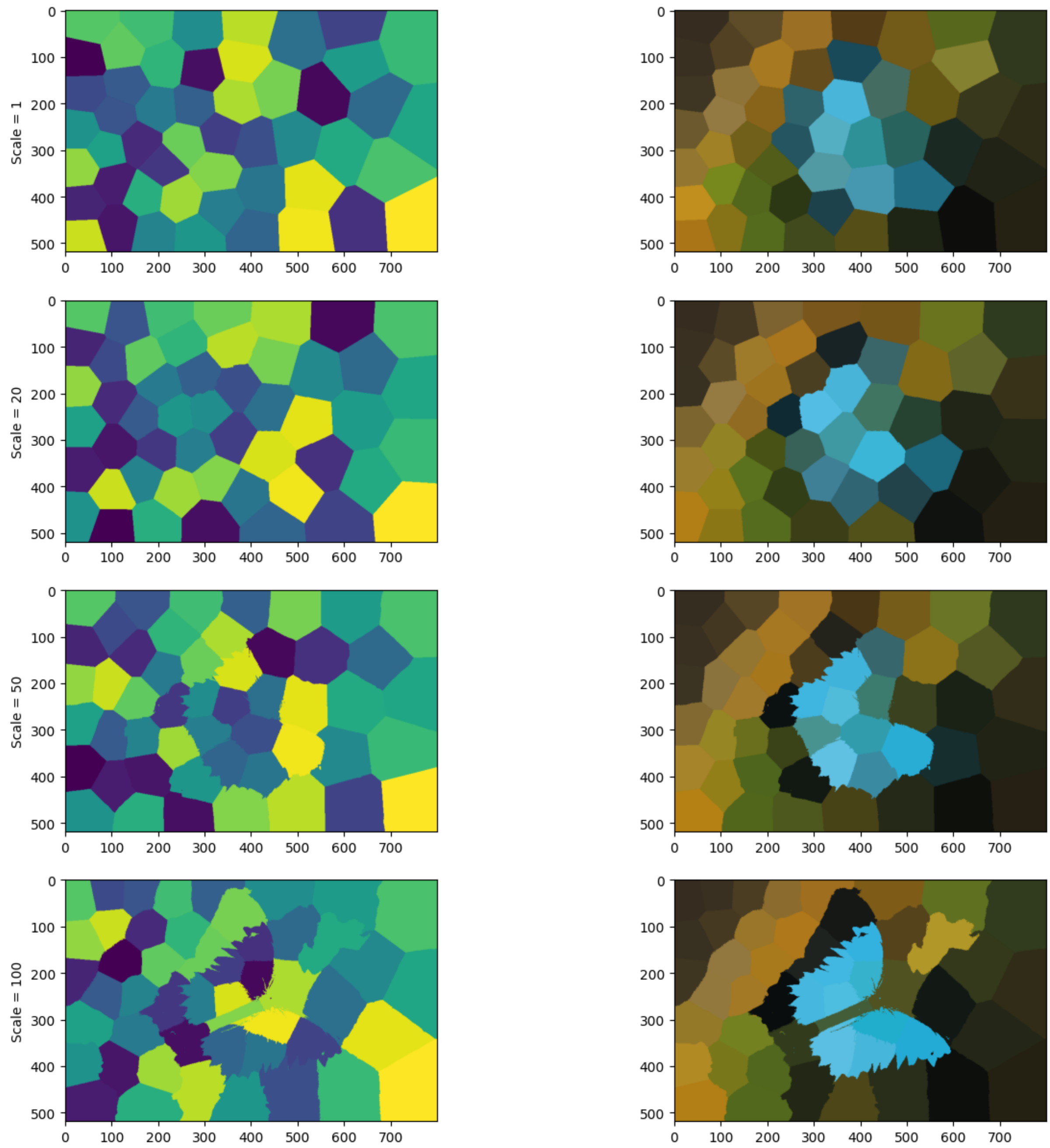


Figure 2: Superpixel example output

Task 4: Discussion

Write a brief report (max 600 words) discussing explaining your implementation and results. Explain your reasoning by referencing the intermediate result figures you generated.