

CSC8628 Image Informatics Project

Abdullah Turki H Alshadadi
School of Computing
Newcastle University
Newcastle upon Tyne, United Kingdom
A.T.H.Alshadadi2@newcastle.ac.uk
190582184

I. INTRODUCTION

This image informatics project aims to develop a classical segmentation method dedicated to accurately segmenting food items within the FoodSeg103 dataset. The segmentation of food holds significant relevance for health-related applications, specifically in the estimation of food calories and nutrients. Hence, the implementation of an accurate methodology is important for achieving reliable results in this context [1].

II. ALGORITHM DESCRIPTION

A. Overview of Region Adjacency Graphs (RAGs)

Region Adjacency Graphs (RAGs) are a method used in image segmentation, where regions of an image are represented as nodes, and edges connect adjacent regions. My implementation is based on RAGs, leveraging the idea that neighbouring regions are likely to belong to the same object [2] [3].

B. 2.2 Image Resizing and Interpolation

To maintain consistency, I resized the images to 256x256 using the nearest neighbour interpolation method. This choice not only reduces computational power but also preserves the original colour values in the image.

C. SLIC Zero Segmentation

A key aspect of the algorithm involves the application of SLIC zero, a subset of the Superpixel-based Linear Index of Connectivity (SLIC) algorithm. SLIC zero's adaptive selection of the compactness parameter enables the creation of superpixels without compromising computational efficiency. These superpixels, generated through a carefully chosen segmentation method, form the basis for subsequent steps in the algorithm. The decision to use SLIC zero reflects a balance between computational efficiency and segmentation accuracy [4]. Furthermore, in the function in **scikit-image** I have set a Gaussian smoothing kernel to smooth the colours so that the next method could take advantage of similar colours [8].

D. RAGs Computation using Mean Colour Method

The superpixels generated by SLIC zero were used to compute Region Adjacency Graphs (RAGs) using the mean colour method. This approach enhances the representation of relationships between adjacent regions by calculating the weight between two adjacent regions representing how similar two regions are in the RAGs [2].

E. Normalised Graph Cut

The segmentation process culminates in the application of Normalised Graph Cut to the computed Region Adjacency Graphs. This step plays a crucial role in refining the segmentation results, ensuring the algorithm's capability to more accurately identify and delineate individual food items within the images [2].

III. LIBRARIES/FUNCTIONS USED

A. Libraries Utilized

The success of the algorithm is underpinned by the strategic use of various libraries:

- **Keras:** Leveraging the `keras.utils.image_dataset_from_directory()` function streamlines the process of reading and resizing the dataset [5].
- **TensorFlow:** The integration of TensorFlow is pivotal in converting the `tf.data.Dataset` to NumPy arrays using the `as_numpy_iterator()` method [6].
- **NumPy:** As a fundamental numerical library, NumPy plays a central role in representing images as arrays [7].
- **scikit-image:** The rich functionalities offered by scikit-image are harnessed for various image processing tasks [8].
- **Matplotlib:** Serving as a versatile plotting library, Matplotlib aids in visually presenting images and critical figures [9].

IV. RESULTS PRESENTATION

A. Dataset Overview

The examination of a specific image, `00000048.jpg` Fig.2, from the dataset serves as a testament to the algorithm's proficiency in accurately segmenting food items. However, a nuanced limitation emerges in the absence of a background mask. This absence leads to the segmentation of the background into sections, even when the background shares the same colour. This nuanced challenge prompts a closer look at background segmentation intricacies.

B. Challenges in Background Masking

Background masking proves to be a formidable challenge due to the diverse colour schemes inherent in different images. For instance, the image `00001977.jpg` Fig.7 presents a

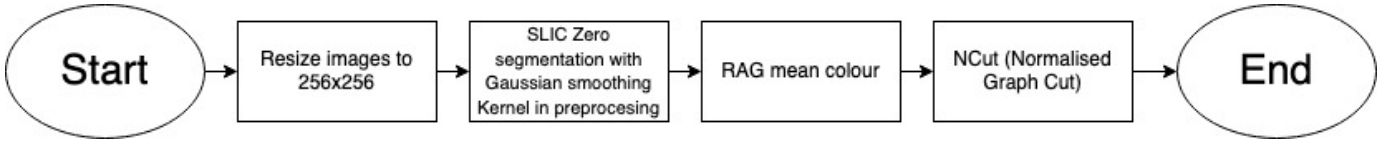


Fig. 1. Flowchart of the implementation

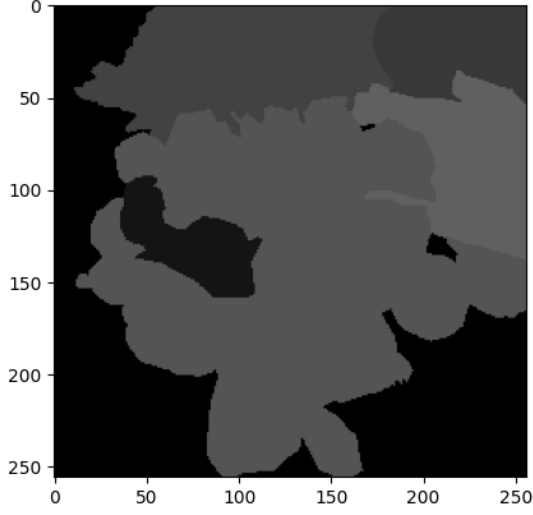


Fig. 2. 00000048 . jpg Ground truth image

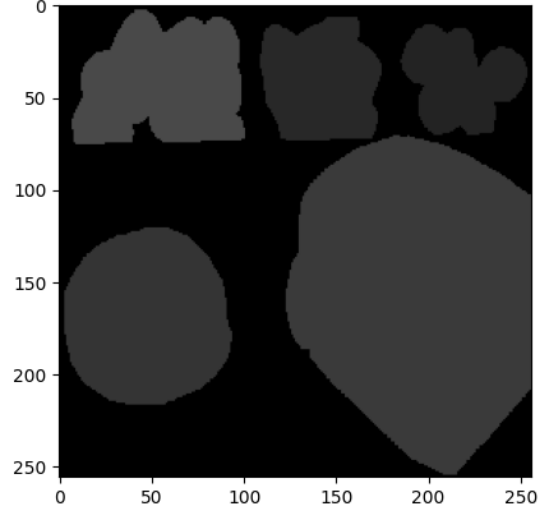


Fig. 4. 00001977 . jpg My own implementation results

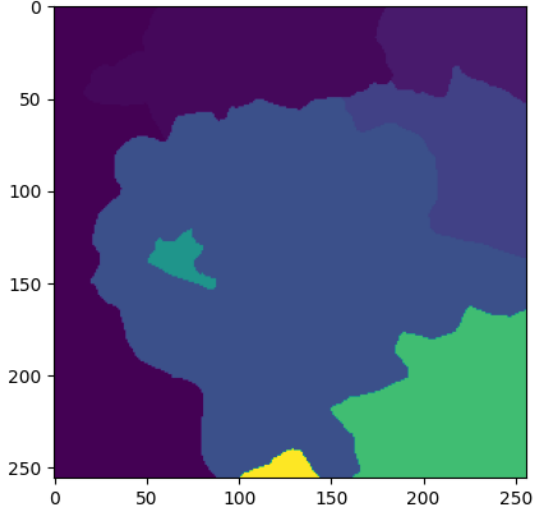


Fig. 3. 00000048 . jpg My own implementation results

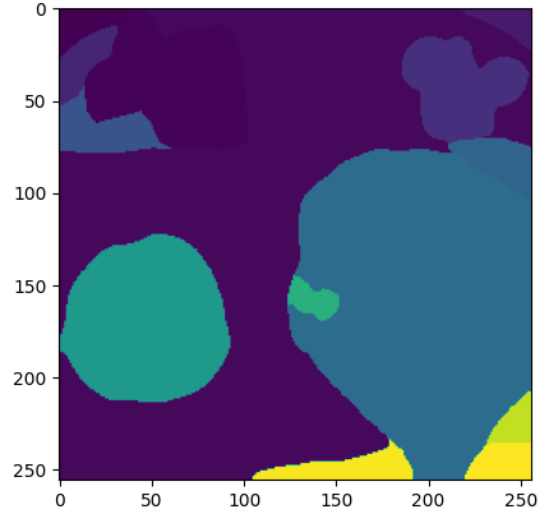


Fig. 5. 00001977 . jpg My own implementation results

scenario where a green plate sits on a white table, adding complexity to the masking process. The presence of shadows further exacerbates the difficulty in achieving accurate background removal. These challenges underscore the need for robust methods capable of overcoming the intricacies of background masking in diverse and dynamic scenarios.

V. KEY FINDINGS AND DISCUSSIONS

A. Strengths and Limitations

The algorithm showcases its strengths in effectively segmenting food items, as demonstrated by the successful segmentation of image 00000048 . jpg Fig.3. However, the absence of a background mask introduces challenges, leading to the unnecessary segmentation of the background particularly when the food extends into the background. This limitation

accentuates the necessity for continued refinement in background removal techniques to enhance the overall accuracy of the segmentation algorithm.

B. Future Considerations

Looking ahead, future endeavours will concentrate on the development of an adaptive masking method. This aims to address the identified limitations by enhancing background removal. The objective is to enable the segmentation algorithm to focus on isolating food items without unnecessary interference from the background. This future trajectory holds promise for significant improvements in the algorithm's overall performance and accuracy.

VI. CONCLUSIONS

In summary, the implemented algorithm demonstrates proficiency in segmenting food items but encounters challenges in background segmentation, particularly with varying colours and shadows. Future endeavours will focus on developing an adaptive masking method to address this limitation. The aim is to create a masking approach that dynamically removes background interference, allowing the segmentation algorithm to focus on accurately identifying and segmenting food items without unnecessary divisions caused by the background.

REFERENCES

- [1] X. Wu, X. Fu, Y. Liu, E.-P. Lim, S. C. Hoi, and Q. Sun, "A large-scale benchmark for food image segmentation," in *Proceedings of the 29th ACM International Conference on Multimedia*, ser. MM '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 506–515. [Online]. Available: <https://doi.org/10.1145/3474085.3475201>
- [2] A. Tremeau and P. Colantoni, "Regions adjacency graph applied to color image segmentation," *IEEE Transactions on Image Processing*, vol. 9, no. 4, pp. 735–744, 2000.
- [3] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, 2000.
- [4] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "Slic superpixels compared to state-of-the-art superpixel methods," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 11, pp. 2274–2282, 2012.
- [5] F. Chollet *et al.*, "Keras," <https://keras.io>, 2015.
- [6] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: <https://www.tensorflow.org/>
- [7] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant, "Array programming with NumPy," *Nature*, vol. 585, no. 7825, pp. 357–362, Sep. 2020. [Online]. Available: <https://doi.org/10.1038/s41586-020-2649-2>
- [8] S. van der Walt, J. L. Schönberger, J. Nunez-Iglesias, F. Boulogne, J. D. Warner, N. Yager, E. Goullart, T. Yu, and the scikit-image contributors, "scikit-image: image processing in Python," *PeerJ*, vol. 2, p. e453, 6 2014. [Online]. Available: <https://doi.org/10.7717/peerj.453>
- [9] J. D. Hunter, "Matplotlib: A 2d graphics environment," *Computing in Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007.

APPENDIX

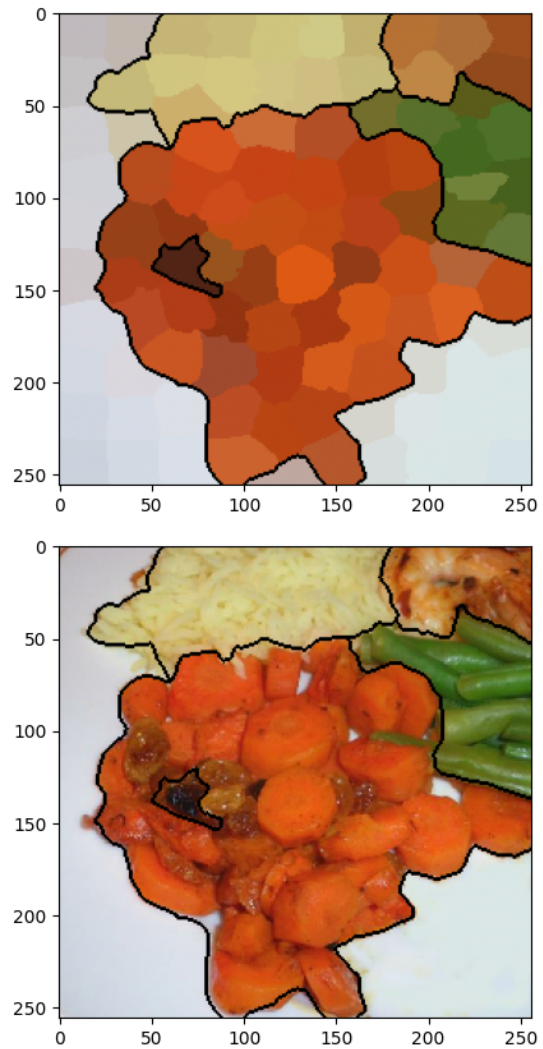


Fig. 6. 00000048.jpg Further visualisation of my implementation results

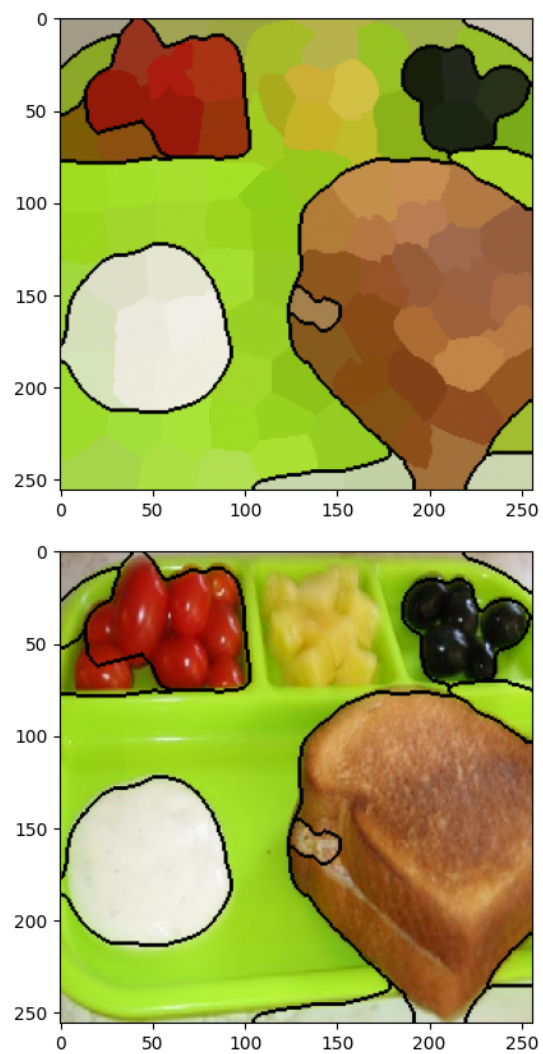


Fig. 7. 00000048.jpg Further visualisation of my implementation results