Advanced R Final Project Proposal: SpecAI.Seg

Scout Jarman, Zac Rios, Eric Aikens, Kyle Steinicke

Introduction

Hyperspectral images (HSI) are high dimensional images, typical taken remotely via satellite or airplane, which contain hundreds of channels of information. This is in contrast to regular color (RGB) images which contain just 3 channels of information. Due to the information richness of HSI, they can be used for many tasks across agriculture, national security, and science. HSI have several practical difficulties, the first is the curse of dimensionality due to HSI having between 100-300 channels, making classification and differentiation difficult. The second is the increasing spatial resolution of images, requiring increased computational resources to process images.

Many tasks, such as image classification, have historically been done on a per-pixel basis. For example, to identify the objects or materials present in an image, one may apply a per-pixel classifier to classify each individual pixel in the image. The downside to this approach is that there is no spatial information included from the image, as each pixel is treated independently. One way to reincorporate this spatial information, is to use a process called image segmentation. Image segmentation is the process of breaking an image into groups of similar pixels. Not only does this help include spatial information, it also helps reduce processing resources, as each segment can be summarized typically using the average of all pixels in the segment.

There are many potential algorithms for image segmentation, though there are some additional difficulties when applying these to HSI. One popular algorithm is the Watershed segmentation algorithm. This algorithm works by find the gradient of an image, then applying a simulated flooding to determine regions of similar pixels. The idea is that if nearby pixels have low gradient, then they are similar to each other, and can be grouped into a segment. To calculate this gradient for HSI one can use the Robust Color Morphological Gradient (RCMG) (Tarabalka *et al.*, 2010).

The most popular software for HSI segmentation is `eCogntition` (Kotaridis & Lazaridou, 2021), so there are not many open source packages for HSI segmentation in R or Python. In collaboration with Los Alamos National Laboratory, Scout Jarman created the `SpecAI.Seg` package in Python to facilitate HSI segmentation in Python (Jarman, 2022). This package provides an easy frame work for loading in a set of standard HSI datasets, as well as the ability to load in custom datasets. These datasets include Indian Pines, Salinas, Pavia University, Pavia Center, Botswana, and Kenedy Space Center (Grana *et al.*, 2011). Additional, this package has several segmentation algorithms in addition to Watershed, such as Slic, Quickshift, and Felzenszwalb. There is also plotting functionality using `matplotlib`.

Proposed Package

`SpecAI.Seg` is, to the best of our knowledge, the first open source Python package specifically for HSI segmentation. Because we want the most amount of researchers to be able to use these HSI segmentation tools, we wish to recreate the `SpecAI.Seg` package in R. This would also be a novel contribution to the R community because though there are package to load in HSI into R, there are no package for doing HSI segmentation in R. The existing R packages for working with HSI are `raster`, `rhdf5`, and `rgdal` Bivand *et al.* (2023). However these package don't offer segmentation functionality, and other packages such as `itcSegment` are too specialized for general HSI segmentation (Dalponte, 2018). Furthermore, because many of the popular HSI datasets are used in Python, they are store as simple `.mat` files, thus using proper spatial statistics packages are unnecessary for our proposed package.

Our proposed package for this final project will be to port over the basic functionality of the `SpecAI.Seg` package into R, using as little Python code as reasonable. Our package will offer three main components:

1. Data loading.
2. Watershed segmentation.
3. Basic plotting.

These three components will be designed to work smoothly with each other by the creation of custom S3 classes and generics. Having such an R package would greatly improve the ability for users to do HSI segmentation in R because there is currently no easy way (i.e. a package) to do such tasks. Additionally, by using S3 classes and generics for plotting, we will take away the tedious work of tracking the various parts involved in HSI segmentation, and make it easy to load, segment, and plot the most popular HSI images.

Challenges

There are a few potential challenges will need to be handled. First, data loading. The standard datasets can be downloaded from the web, so it is possible that if this website were to be unavailable, then data would not be able to be downloaded. It could be possible to have the processed data saved and packaged with the package data, however HSI can be quite large, so they will likely be too large to store. We will experiment with the possibility of storing the formatted HSI data with the package, but will likely need to rely on having the website to download the images locally.

Second is the calculation of the RCMG for Watershed segmentation. This gradient calculation involves calculating the pairwise distance matrix for a neighborhood of pixels around each pixels, meaning there are a lot of calculations. In `SpecAI.Seg` uses multiprocessing to speed up this calculation. We could include Python code for calculating this gradient, and use the reticulate package in R. However we wish to use as little python as possible. Instead we will use RCPP to write an efficient single threaded version of the RCMG.

Third is the plotting of the segment boundaries. This is easily accomplished by using the `scikit-image` function `mark_boundaries` in Python. However, as we have mentioned, we wish to reduce the amount of Python code ideally to zero. Thus we will investigate an R implementation of this function, and if all else fails, we can use reticulate.

Interest

This package will offer some topics that may be of interest to other students in this class. First, is how one can load in data downloaded from the internet. Second, is the application of RCPP to speed up a long calculation. Third, if needs be, how to reticulate and Python inside of R. Fourth, if any students have an interest in image processing, then this package is an interesting application of image segmentation to a high-dimensional image format.

References

Bivand R., Keitt T. & Rowlingson B. (2023). Rgdal: Bindings for the 'geospatial' data abstraction library. https://CRAN.R-project.org/package=rgdal.

Dalponte M. (2018). itcSegment: Individual tree crowns segmentation. https://CRAN.R-project.org/package=itcSegment.

Fischer B., Smith M. & Pau G. (2022). rhdf5: R interface to HDF5. https://github.com/grimbough/rhdf5.
Grana M., Veganzons M. & Ayerdi B. (2011). Hyperspectral remote sensing scenes. In: Hyperspectral Remote Sensing Scenes - Grupo de Inteligencia Computacional (GIC). Grupo De Inteligencia Computacional. https://www.ehu.eus/ccwintco/index.php?title=Hyperspectral_Remote_Sensing_Scenes.
Hijmans R.J. (2023). Raster: Geographic data analysis and modeling. https://CRAN.R-project.org/package=raster.
Jarman S. (2022). SpecAI.seg. https://pypi.org/project/SpecAI.Seg/.
Kotaridis I. & Lazaridou M. (2021). Remote sensing image segmentation advances: A meta-analysis. ISPRS Journal of Photogrammetry and Remote Sensing 173: 309–322.
Tarabalka Y., Chanussot J. & Benediktsson J.A. (2010). Segmentation and classification of hyperspectral images using watershed transformation. Pattern Recognition 43 (7): 2367–2379.