Computer Vision for Automated Planetary Geologic Mapping

Ryan Anderson, USGS Flagstaff

Project Description:
The goal of this project is to develop a computer program that can partially automate the production of geologic maps, using multiple co-registered orbital data sets. The program would allow human users to identify characteristic terrain types, feed "training" examples of the terrain or feature of interest (as seen in multiple different data sets) to the algorithm, and then map that characteristic terrain across an entire set of images. The work would expand on previous similar efforts by using multiple different data sets at once to identify and map features. For example, rather than relying purely on morphology, a thermal inertia and/or hyper-spectral imaging data set could also be provided as input. Efforts for this project would likely be focused on Mars data, given the relative lack of complicating factors such as vegetation and clouds.

Knowledge, skills, and expertise:
This project will require programming skill and knowledge of machine learning and image recognition algorithms such as convolution neural networks, support vector machines, etc. It will also require familiarity with georeferencing, visualizing, and interacting with orbital data sets.

Equipment Requirements:
No special equipment required. Students will need access to computers with a programming language of their choice. Python is preferred, but if an alternative (or additional) language is required for performance reasons, that is acceptable.

Deliverables:
The result of this project will be a program (with source code available in a Github repository) that allows users to load and georeference multiple Mars orbital datasets, interactively define terrain types and features, and automatically map similar features or terrains across multiple input images. The program should also be able to assess how well the algorithm performed (e.g. by comparing with a human-generated map). A GUI is not required, but the program should be developed with ease-of-use in mind. Along with the program, there should be full documentation and a write-up justifying the choice of algorithm(s) and other decisions made during development.