9/4/2019

What I’ve been working on is essentially preparing a training dataset for a future convolutional neural network (and a feed-forward neural net as a control / standard of comparison), and maybe even a Region Convolutional Neural Net (although less likely) as well. For every image in the dataset, I’m annotating them with labels such as “structural damage”, “flooded”, and “lava”. This dataset will be used when I’m training my network to perform multi-label image classification. The way I plan on storing the images with annotation is with classes at the moment, where each instance has an image and its classified labels as its two attributes. I still have yet to talk to Connor if this is the intended way to go. For 9/4/2019 though, I simply looked for resources that could assist with the annotation preprocessing for the training dataset. [https://www.ngdc.noaa.gov/hazardimages/#](https://www.ngdc.noaa.gov/hazardimages/#/) is definitely accessible, but <https://www.digitalglobe.com/ecosystem/open-data> is not, I’ll explain why at a later time.

9/5/2019

My top resource for now though is this: <https://www.ngdc.noaa.gov/hazardimages/#/> because it is the only source where I was able to actually locate the images. I thought this source looked promising too: <https://www.digitalglobe.com/ecosystem/open-data> but once you download from one of the links on there, you get all kinds of weird files with different extension paths, such as “.shp”. However, I’m familiar with “.shp” files and I know they provide dataframes for geospatial data, not actual images. I’m still considering opening the ``.shp” files with Python, just to see if there’s any relevant / useful data for my training dataset. There were “.jpg” images but all I got were abstract mappings, and not easily recognizable disaster images. I’m attempting to extract all the images rather than having to manually save each one via a scrapping tool such as this <https://pypi.org/project/ImageScraper/>

Another concern I’m attempting to address through research is to figure out a way to map each image to its annotations. I was considering using Python classes, but I realized that’s probably not the best way. I know there are other image annotation tools, but they all seem to use bounding boxes. While bounding boxes may be nice, I actually did not want to use them because A) I’m not concerned with where a computer can detect damage in an image, but rather that the computer can detect in the first place and B) if I have to add bounding boxes in addition to labels for my image dataset preprocessing, that would be much longer and tedious rather than the already-amount-of-tediousness I have to go through for labeling the images alone. I decide that if I can’t find one by the end of the weekend, then I might as well stick with an annotation tool that uses bounding boxes for the training dataset.

And well, of course, since this project was inspired my Beaver Works, why not look at the annotation app and see how the images are being crowdsourced? Well, the thing is, when I first looked at it, I was pretty concerned that I wouldn’t be able to understand any of the code. However, I did not let that get in the way. No matter how little I may know about web dev, fundamentally all code is the same (or at least I hoped so). <https://github.com/mit-ll/PSIAP-CAP-Annotation/tree/master/civil-air-patrol-analysis-front-end> <- that’s the link for the web I’m talking about. There seems to be a web crawler on there (image scrapper), and I thought it might be a nice idea to google image scrapers.