SE 3XA3
Samuel Crawford - crawfs1
Joshua Guinness - guinnesj
Nicholas Mari - marin
January 21, 2020

Project Proposal

Team Name

CAS Dream Team

Team Members

Samuel Crawford - crawfs1
Joshua Guinness - guinnesj
Nicholas Mari - marin

Original Project Name

The original project is called google-images-download and it is hosted by GitHub user @hardikvasa.

Software Purpose

The purpose of this software is to search Google images given a keyword and download a number of images to a folder on the user's computer. This could be used to find a large amount training images for an image classifier in a short amount of time.

Software Scope

The scope of our improved software is pretty limited. It involves searching Google, getting command/GUI inputs, analyzing metadata, and file management. The scope of the project can be expanded as we generate new ideas and possible features during the design process.

URL for Original Project

The original project can be found at https://github.com/hardikvasa/google-images-download.

Any specialized hardware requirements?

No. It is a purely software-based product.

Any software license required that McMaster does not own?

No.

Programming Language

The original program is written in Python, and we intend to keep this language. One of the main reasons for this is because our group has quite a bit of experience in it, ensuring that more time can be spent on documentation and writing good quality code rather than learning a new language. Python supports modules rather well, which we can use to facilitate proper software design compared to the monolithic script that exists right now. Further, Python has a variety of libraries that can easily installed with pip and utilized.

Is the Programming Language Feasible for your Team?

We all have an extensive amount of experience in Python, so this will be achievable.

Is the domain knowledge understandable within one term?

The domain knowledge required for this project is web scraping, command line interface script making, and a little bit of web knowledge. This is feasible to understand within a term and implement it more efficiently and professionally.

Number of Lines of Code

Not including supplementary files, like licenses or configuration files, there are about one thousand source lines of code in the original project.

License

This project uses an MIT license.

License allows public redevelopment?

Yes. Part of the MIT license is allowing the public use, copying, and modification of the code, provided that the MIT license is preserved and kept with the code, and that the software is provided in its current state with no warranty, and any problems, whether by misuse or errors in the code, is not the liability of the developers.

Can you compile the original projects source code?

Yes, we can compile and run this original source code.

What would be some test cases for the existing software?

To test the functionality of the existing software, a possible test case would be to ensure the correct amount of images have been downloaded by the program. For example if a limit of 20 images is specified, only 20 images should be found in the output folder. Additional test cases can be added to ensure that the images are downloaded to the correct location.

What changes do you intend to make to the project?

We intend to make the following main chances to the project:

1. Split the monolithic file into multiple logical modules, to better incorporate the software principle of encapsulation.

2. Some files aren't able to be downloaded by the software. An example is the image found here: https://vignette.wikia.nocookie.net/mspaintadventures/images/5/5b/Trolls_looking_at_green_sun.png/revision/latest/scale-to-width-down/340?cb=20180118110537, which is hosted on a wiki page and has some modifiers in the URL (eg. "revision" and "latest"). With our implementation, we can try stripping the URL to its most basic version (ie. https://vignette.wikia.nocookie.net/mspaintadventures/images/5/5b/Trolls_looking_at_green_sun.png), and trying to download that stripped version instead.

3. Add the additional functionalities of whitelist and blacklist modes where the software will only download images from sources specified by the user (whitelist) or download images from all sources except those specified by the user (blacklist). This will be useful for limiting results from domain-specific or approved websites, and blocking results from known malicious or unwanted sites.

4. The addition of a simple GUI to make the software easier to navigate for users unfamiliar with command line arguments. Use of the GUI should be optional, and the software should have the same amount of functionality from the command line as it does in the GUI.

5. The functionality for users to be able to preview images that will be downloaded before downloading them. This feature should be optional to the user so that they can skip the preview if they are downloading a large selection of images.

6. Allow the user to pass a bit depth argument to only find images with the specified bit depth. This would be useful if using this program to harvest images to use with an image processor that is dependant on a specific bit depth (like our group did at DeltaHacks V and lost because they gave us an image to grade us on that had a different bit depth that our program couldn't handle).