
Google Images Downloader

<https://gitlab.cas.mcmaster.ca/guinnessj/google-images-downloader>



Nicholas Mari
mari-nicholas

Software Tester

Nintendo 3DS Friend Code:
3196-5807-6182



Joshua Guinness
joshuaguinness

Project Manager

Instagram:
@joshua_guinness



Sam Crawford
samm82

Git Specialist


YouTube:
SamsUndertale


Agenda

1. Introduction and the “Why”
2. Development Process
3. Demonstration
4. Conclusion and Next Steps
5. Questions


Introduction

- Reimplementation of:


 [hardikvasa](#) / [google-images-download](#)

 Watch ▾


189


 Star


6.6k


 Fork


1.5k


 Code


 Issues **65**


 Pull requests **19**

 Actions

 Projects **0**

 Wiki

 Security

 Insights

Python Script to download hundreds of images from 'Google Images'. It is a ready-to-run code!

python

python-script

image-download

google-images

image-processing

color-filter

image-dataset

image-database

image-search

image-scraper


image-gallery


terminal


command-line


command-line-tool


download-images


 **156** commits

 **3** branches

 **0** packages

 **0** releases

 **34** contributors

 MIT

What is the problem being solved?

Machine Learning:

- If doing any image recognition ML, this tool is any easy way to download similar images on your computer

Arts:

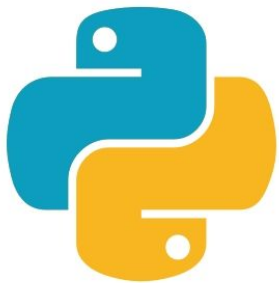
- Arts projects that require lots of similar images e.g. collages
- Inspiration

Start of Development Process

- ERROR
- Google changed their implementation of how images are stored
 - Image URLs are resolved dynamically so downloading the entire HTML page as in the original code won't work
- Previous code wasn't modularized or tested

Implementation Details

- Coded in Python as our group was comfortable with it and because it is a good scripting language
 - This was also the language of the original implementation



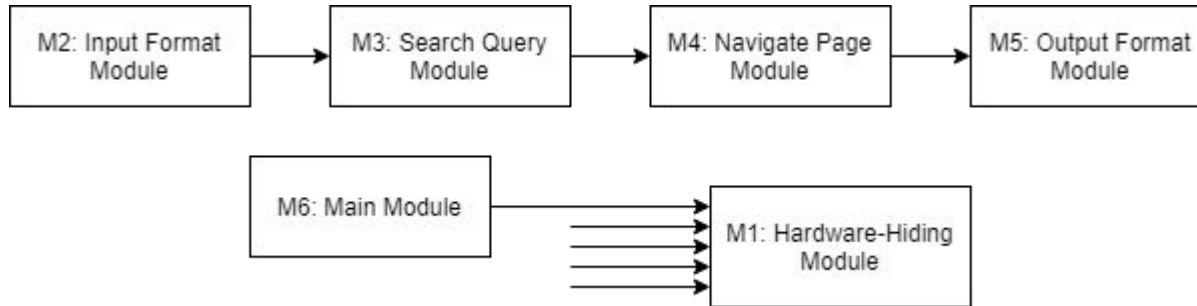
Selenium



- Due to problem outlined last slide, would need to use browser automation via selenium to get the image URLs

Architecture

- Data Flow Architecture because it is a collection of scripts
 - Batch Sequential!



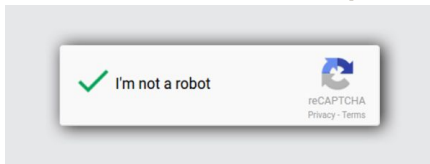
Demonstration

1. Color and Image Size
2. Filetype
3. Blacklist and Whitelist
4. Keyword From File
5. Move to Server (Time Permitting)

Possible Next Steps

Known Bugs:

- If Google search query returns # images < limit, the script will run forever
- If trying to download > 55 images



RIP :(

Improvements:

- Better error handling, especially around moving images to a server
- End User Readiness
 - Stronger documentation and user guide instructions
 - Ready to ship and install on user systems with 'pip'

Conclusion

- Key functionality of original plus lots of additional features implemented
- Learned about Selenium, Python, and web scraping
- Had a good time and grew closer as friends :))

Questions

Ask us about our implementation process, code, functionality, why Sam has a concussion, or to see our “move to server” demo