Development of automatic Landsat data download engine

K A Pradono[1], Y D Safitri[1], B S Adhitama[1], Y F Hestrio[1], M Soleh[1], H Gunawan[1], N Widijatmiko[1], W Sunarmodo[1]

[1]Pusat Teknologi dan Data Penginderaan Jauh LAPAN

Email : kuncoro.adi@lapan.go.id

Abstract. Remote sensing data usually collected by direct acquisition using ground station. But, limitations on acquisition often leads to missing data (especially the old data). United States Geological Survey (USGS), as one of scientific agency and Landsat remote sensing satellite operator, provides access for users to their old to recent remote sensing data. USGS also provides tools to help user to download their data in large or small quantities, but the tools have limitations. The most noticeable limitations are error when downloading and low download speed. At the same time, access and download data in large quantities without tools would takes a lot of time and effort. Hence, an engine to overcome those limitations needs to be developed. By using this engine, users are expected to be able download remote sensing data in large/small quantities automatically, with higher download speed, and less error or corrupt data.

1. Introduction

Remote sensing data is generally collected by direct acquisition via ground stations. However, due to some limitations such as equipment, budget, and development of acquisition tools, often leads to missing data (especially the old data). Nowadays, there are server node networks that provides such data such as United States Geographical Survey (USGS). Landsat data is an example of remote sensing data that is acquired by Pustekdata LAPAN and also provided by USGS.

To download information and data, users must first wait while the browser is open, contact the web server indicated in the URL, then download the appropriate HTML page to the user's computer. Depending on the type of connection and size of the web content being downloaded, the configuration goes down, including handheld devices, multiprocessor systems, microprocessor-based or programmable consumer electronics, minicomputers, mainframe computers, and loading step can take several minutes [1].

USGS provides tools to access and download any available data in their catalogue. The tools are quite useful for users to download their data, but, in fact, there are several limitations faced by users on using the tools. The most noticeable limitations are data error when downloading and low download speed. Therefore, it is necessary to develop an automation engine to overcome those limitations. The engine need to be able to automatically download the data needed by the user, and able to download the data on acceptable download speed and low error rate.

The automation process is done by making an application so that all activities are done automatically using technology to control and do work that is usually done manually, also reducing production costs due to remote monitoring [2] [3]. By using this download engine, it is expected that the user only need to input the required Landsat scene list. The advantage of this engine is hopefully help user or operator to work faster, more efficient, and no corrupt data. One of the benefits that has been felt is to maintain

the Service Level Agreement (SLA) for Landsat-8 data availability. Through this engine it is hoped that data can be collected quickly by users or operator.

2. Methods

The engine use scraping technique which commonly used to extract information. Nowadays, there are quite a lot of research in the field of information extraction from various types of events, content or relationships related to textual data. Such information is usually used for search engines, latest software libraries, technical instructions or dictionaries. A form of information extracted from the text aims to find something new, a new way to solve a problem. Web scraping is a technique commonly used to gather information from a website rather than extracting it manually [4].

The data to be extracted is content on the website which procedure requires human to do data extraction work. The extraction data needed is not in the form of text like most scrape process on a website, but rather a remote sensing image of Landsat satellite. The diagram of the engine can be seen on Figure 1 as follows.

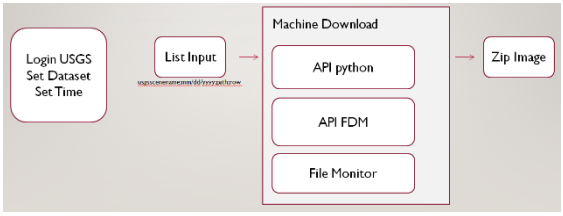


Figure 1. Block diagram of the engine

USGS Login credentials, dataset, and time set is the properties that user has to provide to the engine. "List Input" is an information about list of the data that user needs. Usually the user already has a list of what scenes they need to download. In manual operation case, commonly the user downloads all the data they wanted one by one. Through this engine, the user only required to provide list of data they need to be downloaded, then the rest will be done automatically by the engine. List of the data has many possible ways to have, but the most common way is using "Search Results" feature from the USGS data portal (Figure 2).

Figure 2. "Search Results" feature from the USGS data portal

The automatic download engine consists of 3 modules, namely the main software (Python API), Free Download Manager (FDM) API, and monitoring software (monitor API). On main software, run loop process to input list of scene names user want to download which every scene name contain scene recording time and scene location information, so that at the end of the search session it is ready to be

downloaded. On the main software also done download timing configuration so that the load on the provider's server is not overloaded and the internal network of user maintain stable when downloading. A good website usually prevents users from interacting to quickly, like bots, although security features do not look like they are installed. Scrape and download too much data so the provider's server goes down will give the user a bad effect to be blocked from the IP side and get a warning [5].

Free Download Manager (FDM) API is an open source download accelerator software. FDM has ability to download data using HTTP, HTTPS, FTP, bittorent and metalink. FDM main features are upload manager, site explorer, and HTML spider. Site explorer displays the structure of a web to download files that are needed and the user can fully download the files contained therein [6]. Site explorer capabilities that are used is to move the download process from the browser automatically to the download accelerator.

A computer system such as a process or throughput is a functional part of hardware and software efficiency. Identification of the performance capabilities of a computer system needs to be monitored by executing software that is still part of the system being built [7]. The capabilities to monitor the process when the engine run has been added through the ability to see the data being downloaded as seen on the Figure 3.

Figure 3. Monitoring process of the engine

2.1. Python

Python is an interpreter, a programming language for various purposes, created by Guido van Rossum in 1991 which updated in 2000 with version 2 and updated again in 2018 with version 3. the focus of python is at the level of code readability so it is more human to read. Some of the advantages of Python programming language are comprehensive libraries functionality and support of huge developer community while also being equipped with automatic memory management functions. Python runs on various operating systems such as Windows, Linux, Mac, and so on [8]. Python provides services to application developers to write code quickly. The presence of bytes compiler and supporting libraries increases the performance of applications that use Python [9]. Python can be used freely, even for commercial purposes. Many companies use Python programming language to develop commercial products to provide services. Syntax in Python can be run and written to build applications on various operating systems [10].

Python 3 is the programming language that is used when developing this automatic Landsat data download engine. It is felt that Python is a Programming language which has huge number of libraries. The main library that will be used to support the functionality of the engine is PyAutoGUI library. This library is chosen because it has the feature to do automation based on Graphical User Interface (GUI) work.

2.2. PyAutoGUI

By using this library, command prompt console able to control mouse and keyboard. This automation ability is used to interact with other applications that are connected in download engine design. This library is part of Python so it is compatible with various operation systems such as Windows, Mac, and Linux.

Some main features that are available on PyAutoGUI are mouse cursor driver, automatic click or input on form, keyboard task such as print-screen, display message to user, and automation. To using this library properly, computer need to be trained when interacting to the desired application.

2.3. Free Download Manager

Free Download Manager is free and open source software in the beginning. It is currently a proprietary software but still free to use. Source code is available in version 3.0825 in 2010 with a complete binary

package. Up to version 3.9.7. GNU GPL since version 5 has been revoked so that this software is proprietary but still free to use.
The main features of FDM are :

- Graphical User Interface
- Drag & Drop on Dropbox account
- HTTP and FTP download support
- RTSP/MMS download support
- Multiple download support
- Bittorent file download support
- Resume when download disconnected or stopped
- 3 download accelerator mode, which is light, medium, and high mode
- Remote access capability

As for what is utilized from FDM on the download engine are the ability to accelerate downloads, resume broken download, and the ability to download multiple files at the same time. These three features are critical to the success of the Landsat data download process from USGS. As these features have not been able to facilitate.

3. Implementation result

In the implementation of the download engine, Python used as an intermediary implementation. Python will detect input from the computer interface, where there seems to be an operator who do the work. On supporting these interactions, the engine requires some variables/information such as login credentials, scene location (path-row), scene recording time, and other variables.

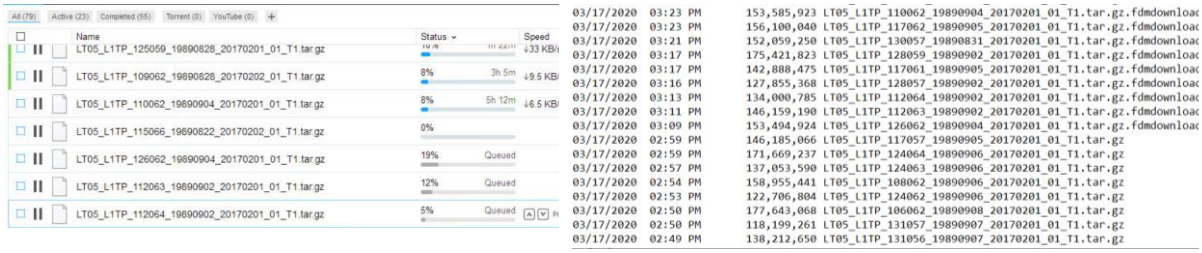
Those variables contain constant variables and repeated variables. Constant variable is set by user defined, and repeated variables will be programmed in Python. The user defined variables are login credentials (username and password), data type, data level, data tier, cloud cover percentage. As for the repeated variables are scene information which are scene location and recorded time, so, to support this variable, the user required to have scene list to input. This list contains scene name (USGS Standard), date according to the format of input data on the USGS web, CCC, path and row. From the list that will be an input to the python, it will be iterated so that it seems there are operator.

The previous explanation is a brief description of how the download engine works. But to be able to automate the download job, some details that have been programmed to software are needed. Those details are software coordinate, download state, and cache clearance. Browser and downloader software coordinate to do GUI automation configuration are needed. Also, the download state configuration has been programmed into the software. The software manages the waiting state for the next step, waiting step to wait for download process and finish step when the process is finished.

3.1. Download performance data

After the system can be implemented properly, further task to do is to ensure the engine runs properly by doing performance calculations. This performance calculation is the results of comparison of data input to successfully downloaded data. In reality there are still download failures due to various technical reasons. Those reasons are bandwidth that suddenly becomes unstable, bugs in the software, disruption of the operating system and other unknown events.

In testing the download process in this automatic download engine, some data from USGS is used including Landsat 5, Landsat 7 and Landsat 8. The amount of data in the list also varies between 50, 100, 200 to 300 data. And the results of the downloads also vary in percentage, start from 100% up to 30%. The time spent by the engine is certainly faster than the tools provided by USGS. The following Figure 4 (a) and (b) are some data that has been taken from the download process.



(a) (b)

Figure 4. (a) Download process from FDM, (b) Download Result

Table 1. USGS tools (BDA) and Download Engine (DE) comparison

Software	Total Data	Success	Failed
BDA	269	196	73
DE	250	232	18
BDA	50	27	23
DE	50	48	2

Table 2. Download time comparison

Software	Total Data	Time (hours)
BDA	269	216
DE	250	14
BDA	50	13
DE	50	2

The scene results of the automatic download engine are compressed data files in particular format. The format that used by USGS is .tar.gz compression standard format. The compressed data files can be extracted with common archive manager software such as 7.zip and WinRar.

The successful extraction process will produces a folder contain geotiff files and .txt format metadata files. This RAW format is needed and utilized by other divisions and possibly other agencies to produce information they wished or required.

The success of the engine is not only measured by the number of files (Table 1 and Table 2) that have been successfully downloaded, but also the success of the extraction of files or downloaded data. Development of extraction process/software is done to support the functionality of the engine as seen on the Figure 5.

The workflow is that all downloaded files will be moved to a specific folder, then all .tar.gz format files in the folder will be listed. Extraction software will work to extract each scene (files) in the list and place it in a folder that named using standard USGS scene naming as seen on the Figure 4 (b).

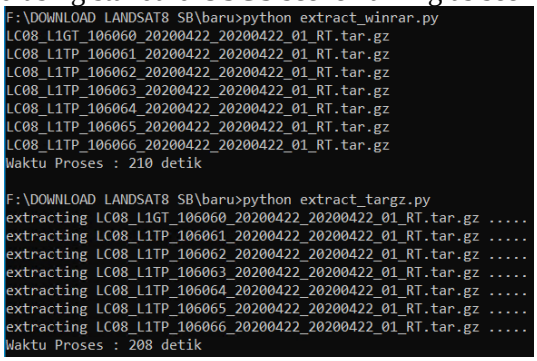


Figure 5. Extract process

From this specific process a comparison to licensed extraction software, WinRar, is done. Same amount of files will be extracted by methods, and then will be checked the file completeness of extraction process in each folder. Process execution times of both methods is also done to compare efficiency of each process. The result is both methods produces exactly same amount of files by their process and the execution time is also almost the same. Therefore, the development of extraction/software will boost the engine functionality and efficiency as add-ons or additional features.

4. Conclusion

From the observation and analysis that has been done in the section before, it can be concluded that the automatic download engine is runs properly and as expected. The efficiency of time and the results of downloads greatly helps the provision of data. Although there are specific requirements or prerequisite needed to operate the automatic download engine. It is hoped that the engine presence helps operational work on providing Landsat data to be easier and better, as the engine is developed to automate work process. In the future several features can be added to the engine to increase engine's functionality such as automatic data extraction or others.

5. Acknowledgements

Authors would like to thank to all of unmentionable colleagues in Pustekdata LAPAN on helping authors directly or indirectly on developing the engine and writing the paper.

6. References

[1] J. Jameossanaie, "Automatic Download of Web Content in Response to an Embedded Link in an Electronic Mail Message," vol. 2, no. 12, 2015.
[2] D. C. Karia, V. Adajania, M. Agrawal, and S. Dandekar, "Embedded Web Server Application Based Automation and Monitoring System," *IEEE*, no. ICSCCN, pp. 634–637, 2011.
[3] Y. D. Safitri, A. S. Nasution, W. Sunarmodo, H. Gunawan, and A. Widipaminto, "Otomatisasi Pemantauan Capaian Service Level Agreement Satelit Landsat-8 di Stasiun Bumi Penginderaan Jauh Rumpin," *Majalah Inderaja*, Jakarta, pp. 10–14, Nov. 2018.
[4] E. Vargiu and M. Urru, "Exploiting web scraping in a collaborative filtering- based approach to web advertising," vol. 2, no. 1, pp. 44–54, 2013, doi: 10.5430/air.v2n1p44.
[5] R. Mitchell, *Web Scraping with Python : Collecting Data from the Modern Web*. 2015.
[6] M. Yasin, M. A. Wahla, and F. Kausar, "Analysis of Free Download Manager for Forensic Artefacts," pp. 59–68, 2010.
[7] M. R. Fortin, Austin, and Tex, "System and Method for Enabling Stripped Object Software Monitoring in a Computer System," no. 19, 1996.
[8] Wikipedia, "Python (programming language)." https://en.wikipedia.org/wiki/Python_(programming_language) (accessed Jul. 17, 2020).
[9] P. Kannan, S. K. Udayakumar, and K. R. Ahmed, "Automation Using Voice Recognition with Python SL4A Script for Android Devices," *IEEE*, no. August, pp. 28–31, 2014.
[10] Advernesia, "Pengertian Bahasa Pemrogaman Python," 2018. https://www.advernesia.com/blog/python/pengertian-bahasa-pemrograman-python-dan-kegunaanya/ (accessed May 30, 2018).