

FIT 3162 Computer Science Project 2

CODE REPORT

GPU Acceleration for Raster Filter Using APARAPI

Group 9

Christine 29392888

Wan Jack Lee 28848551

Zisong Liao 28418107

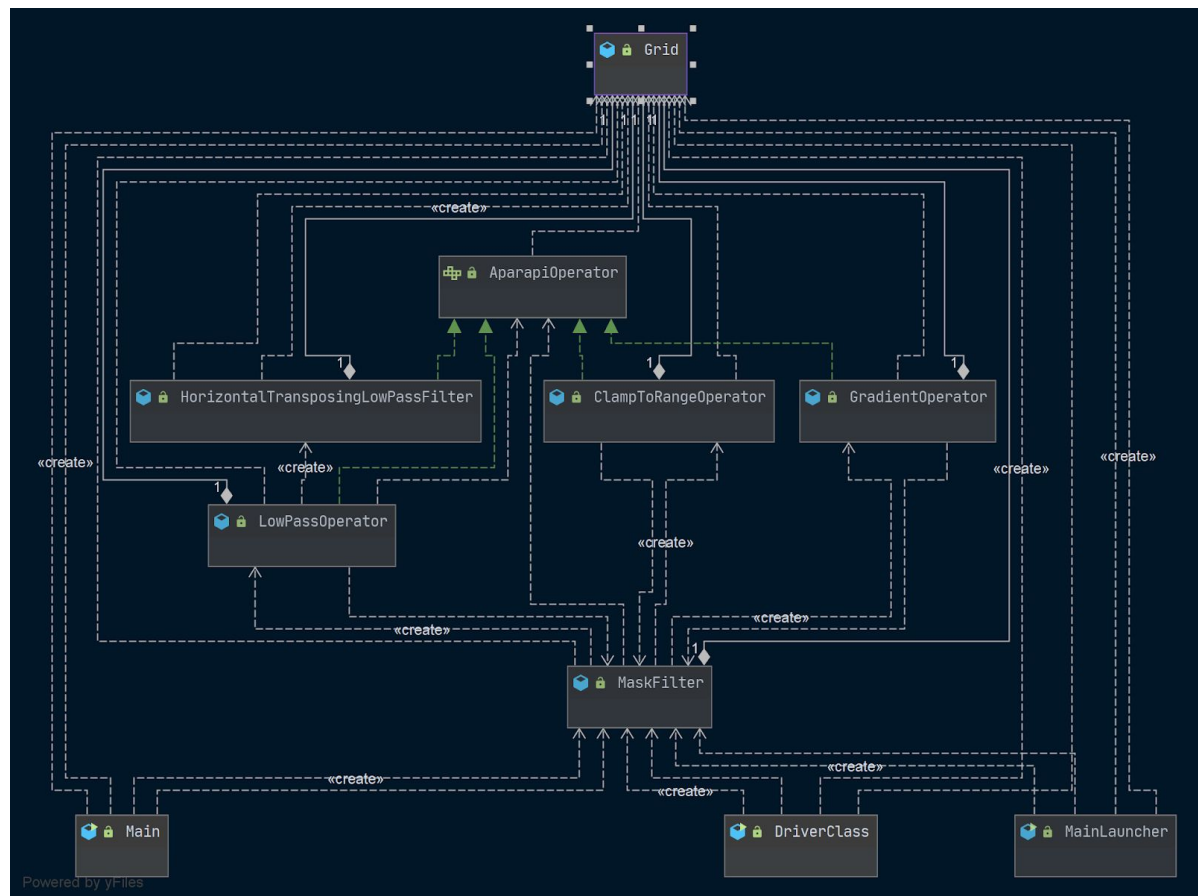
Table of content

Table of content	1
Inspection of code	2
Consideration of software	2
Robustness	2
Scalability	2
Platform and OS independence	2
Limitations of code and future work	2
Basic End User Guide	4
Technical User Guide	7

1. Inspection of code

Here is the link to the repository of the APARAPI-acceleration project:

<https://github.com/OneJackLee/APARAPI-acceleration.git>



This is the class diagram of the overall APARAPI-acceleration package.

As shown in the diagram, the MainLauncher class will be the main class of the APARAPI project and its purpose is to deal with the Input Output operation. Besides that, it also triggered the mask filter algorithm (host the instances of MaskFilter object), generating the raster output, raster effect image, and benchmark file.

The Main class is the original driver class which can be used to link to the Eduard package ProcessLauncher. It is not suggested to be used as some of the functionality needs to be manually modified. Apart from that, it serves the same purpose of the MainLauncher class. Also, the DriverClass is for testing purposes thus it can be ignored.

The Grid object of APARAPI serves the same purpose of the Grid object of Eduard. It hosts the vital attributes and functions.

The AparapiOperator interface has been created in order to achieve polymorphism. It defines the abstract method which to be implemented by its subclasses. It standardized the method that called to execute the APARAPI kernel. The

HorizontalTransposingLowPassFilter (operator), LowPassOperator, ClampToRangeOperator, and GradientOperator are the child classes of AparapiOperator. Each operator serves the same purpose of its counterpart operator in Eduard software.

2. Consideration of software

2.1. Robustness

The acceleration software has ensured the robustness. For instance, the input data has properly checked before applying the mask filter algorithm. If an incorrect data type is given by the user, the software will display the reason for causing the error and exit without crashing. Also, if no input is given, the software will display a warning and exit the software too. Other than that, if there are any exceptions raised during runtime, the main try-and-catch block will catch those exceptions, display the error message and exit the software.

(refer to the [Test Report - Usability Test](#))

2.2. Scalability

The language that used to develop the acceleration software is JAVA, which supports Object-oriented design. Thus the extensibility of the software can be achieved by just creating new classes with the implementation of the algorithms. As the algorithms are modified based on the Mask Filter generator algorithm (retrieved from Eduard software), there is no arbitrary feature that will fail when some certain input is given.

2.3. Platform and OS independence

The software is developed based on JAVA language. Java is a cross-platform language and can run across on Windows, Linux and macOS. The GPU acceleration library: APARAPI will compile its system call to the low-level OpenCL system call during its runtime. Most of the graphic cards support OpenCL, meaning that the OpenCL system call that invoked by APARAPI can be executed in those platforms without any issues. Thus this APARAPI-acceleration software is cross-platform supported.

3. Limitations of code and future work

The current limitation of the software is that the overall program runtime is not optimized compared with Eduard software. The culprit of the longer runtime is induced by the limitation of the APARAPI. As there is no way to access multiple rows of the array at the same time with using APARAPI, thus the algorithm will only be executed in sequential order, which will not fully utilize the benefit of using GPU compute units. Besides that, another reason for pulling the execution speed back is the communication overhead and the workload of the GPU. The communication overhead is caused by moving the data between CPU memory and GPU memory, which is unable to avoid. Besides that, the Eduard Grid

object is not fully compatible with the APARAPI version of Grid object, thus needing to translate the object (casting is not supported).

The future works include replacing(refactor) the APARAPI grid class with Eduard grid class, which requires modification of both Eduard classes and APARAPI classes. Besides that, as the algorithm of horizontal transposing operators needs to access multiple rows at the same time, we can integrate another acceleration method, for instance, OpenMP library can be applied to the operator.

4. Basic End User Guide

The precondition of executing the software is that JAVA Runtime Environment must be installed on the client computer.

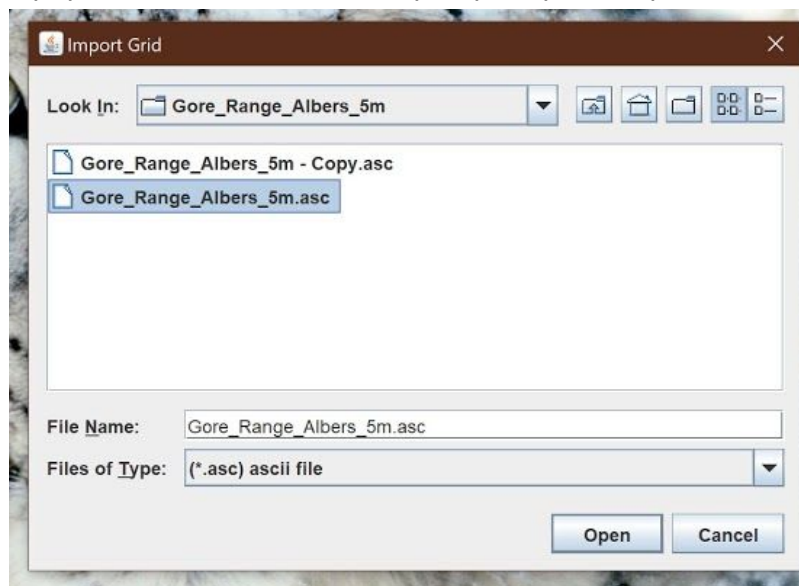
The APARAPI-acceleration software can be downloaded through this link:

[APARAPI-raster_filter.jar](#) . It is a portable software thus no installation needed.

After download the APARAPI-raster.jar file, double click the JAR file to execute the software.

After executing the software,

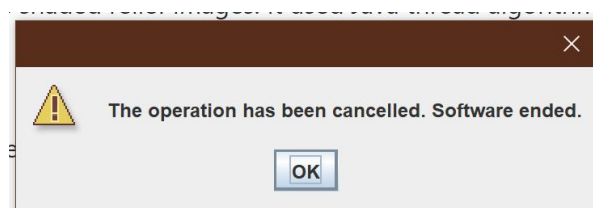
1. A pop out window will shown up to prompt the input raster file *.asc.



- 1.1. If the input file is not supported, an error message will shown up and the software will end itself.

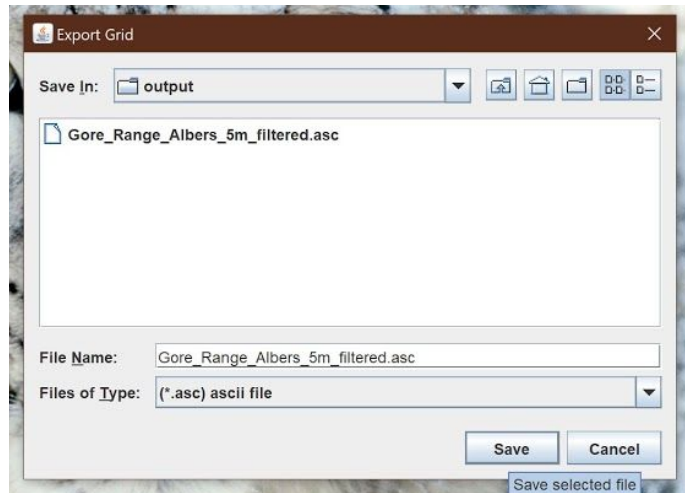


- 1.2. If there is no input file, a warning message will shown up and the software will end itself.

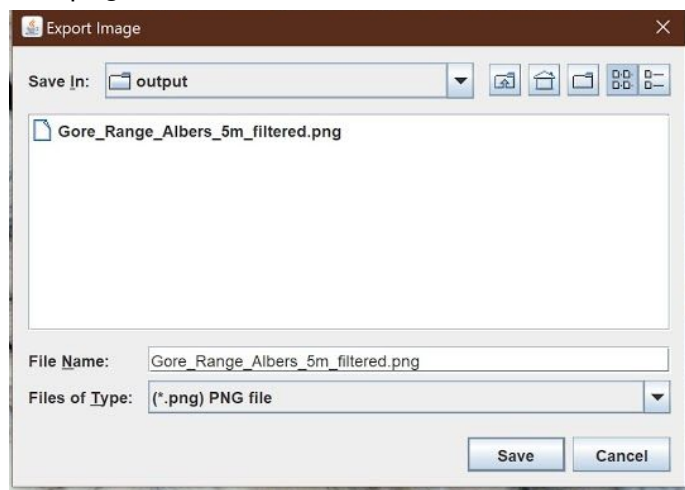


- 1.3. Otherwise, it will proceed to next step.
2. It will take time to process the raster file and generate the mask filter.

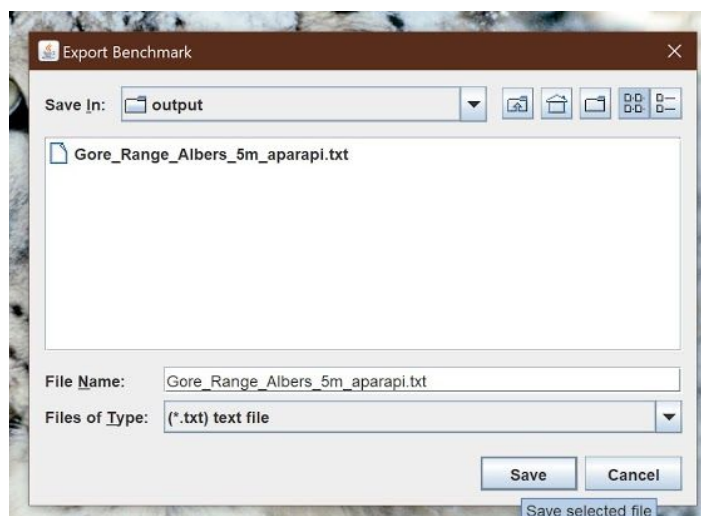
3. A pop out window will shown up to prompt the output location of the filtered raster file *.asc.



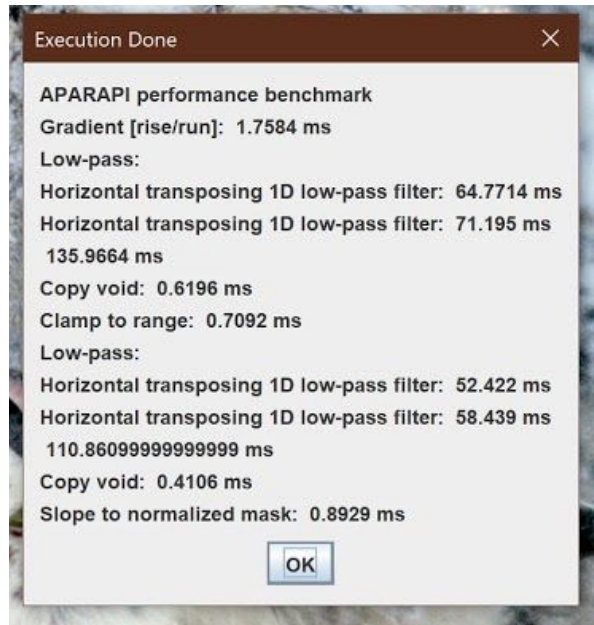
4. A pop out window will shown up to prompt the output location of the filtered image file *.png.



5. A pop out window will shown up to prompt the output location of the performance benchmark file *.txt.



6. Once the exports of those files are done, a done execution message will pop-out.



5. Technical User Guide

The APARAPI-acceleration software requires JAVA development kit 1.8 or above, and Java Runtime Environment to execute the software. To modify the code itself, MAVEN must be installed in the devices.

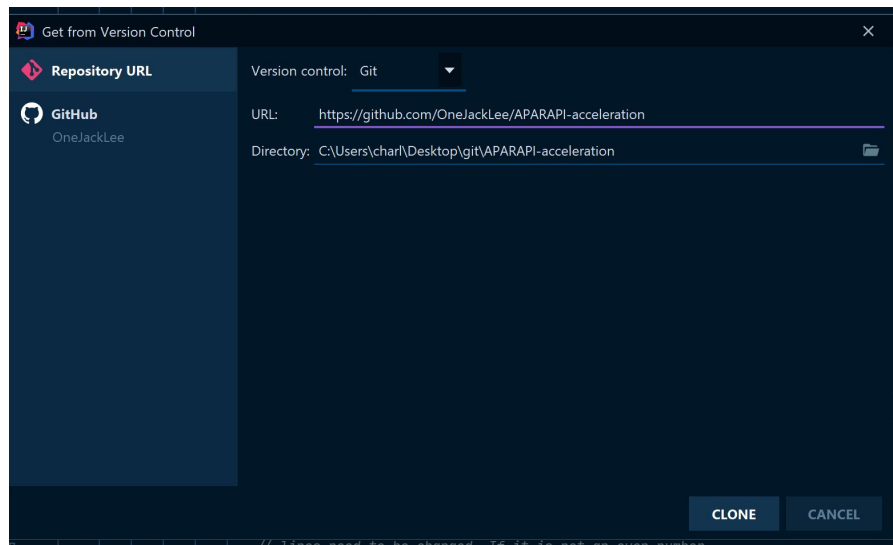
Besides that, there is no minimum requirement of the hardware as all of the devices have either dedicated graphics cards or integrated graphics cards.

The repository of the APARAPI-acceleration can be found here:

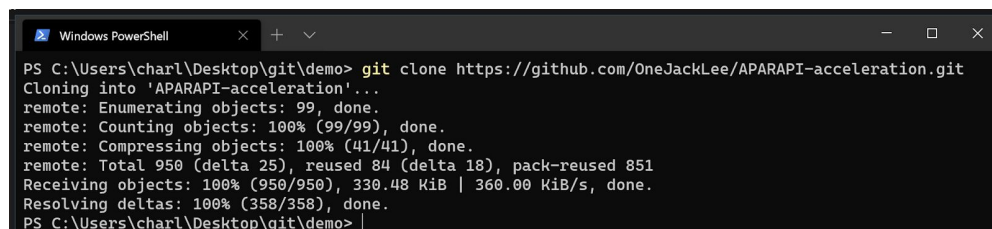
<https://github.com/OneJackLee/APARAPI-acceleration>

There are multiple ways to download the project.

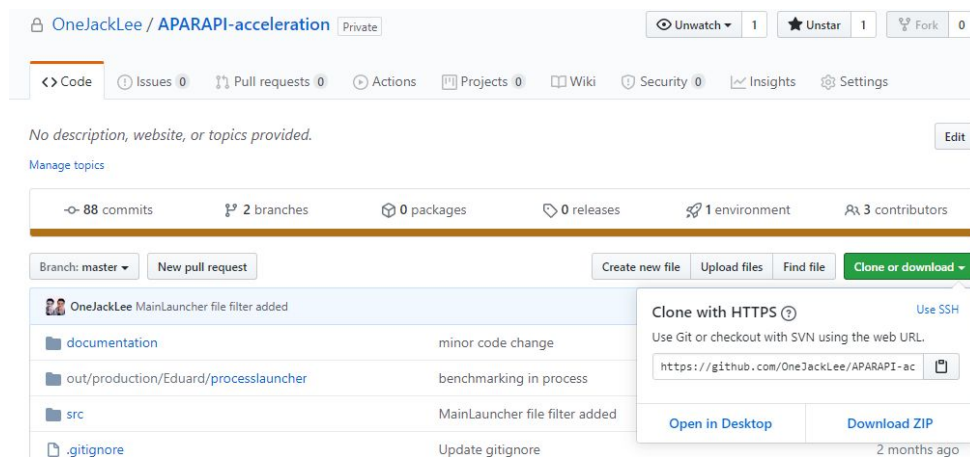
- use any JAVA-supported IDE to clone the project



- open the command prompt or terminal and execute the command `git clone link_of_this_repo` (required git installed)



- directly download this project from gitHub.



Once downloaded the project, to compile the Java code:

1. Open the command prompt or terminal.
2. Navigate to the APARAPI-acceleration parent folder

```
Windows PowerShell
PS C:\Users\charl\Desktop\git\demo> cd .\APARAPI-acceleration\
PS C:\Users\charl\Desktop\git\demo\APARAPI-acceleration> dir

Directory: C:\Users\charl\Desktop\git\demo\APARAPI-acceleration

Mode                LastWriteTime         Length Name
----                -
d-----        6/3/2020   7:56 PM                documentation
d-----        6/3/2020   7:56 PM                out
d-----        6/3/2020   7:56 PM                src
-a-----        6/3/2020   7:56 PM             606 .gitignore
-a-----        6/3/2020   7:56 PM             429 performance.txt
-a-----        6/3/2020   7:56 PM            3420 pom.xml
-a-----        6/3/2020   7:56 PM            1732 README.md
-a-----        6/3/2020   7:56 PM             397 useful_git_command.txt
```

3. Execute the command `mvn package` to generate the JAR executable file.

```
Windows PowerShell
PS C:\Users\charl\Desktop\git\demo> cd .\APARAPI-acceleration\
PS C:\Users\charl\Desktop\git\demo\APARAPI-acceleration> dir

Directory: C:\Users\charl\Desktop\git\demo\APARAPI-acceleration

Mode                LastWriteTime         Length Name
----                -
d-----        6/3/2020   7:56 PM                documentation
d-----        6/3/2020   7:56 PM                out
d-----        6/3/2020   7:56 PM                src
-a-----        6/3/2020   7:56 PM             606 .gitignore
-a-----        6/3/2020   7:56 PM             429 performance.txt
-a-----        6/3/2020   7:56 PM            3420 pom.xml
-a-----        6/3/2020   7:56 PM            1732 README.md
-a-----        6/3/2020   7:56 PM             397 useful_git_command.txt

PS C:\Users\charl\Desktop\git\demo\APARAPI-acceleration> mvn package
[INFO] Scanning for projects...
```

The executable JAR file `APARAPI-acceleration-1.0-SNAPSHOT-jar-with-dependencies.jar` will be located in the `target` folder (in the `APARAPI-acceleration` parent folder). To execute the JAR file, simply double click on it or use the command `java -jar filename` in the command prompt. For execution details, please refer to the [Basic End User Guide](#)

Command Prompt View

```
PS C:\Users\charl\Desktop\git\demo\APARAPI-acceleration> dir

Directory: C:\Users\charl\Desktop\git\demo\APARAPI-acceleration

Mode                LastWriteTime         Length Name
----                -
d-----         6/3/2020   7:56 PM                documentation
d-----         6/3/2020   7:56 PM                out
d-----         6/3/2020   7:56 PM                src
d-----         6/3/2020   7:59 PM                target
-a-----         6/3/2020   7:56 PM             606 .gitignore
-a-----         6/3/2020   7:56 PM             429 performance.txt
-a-----         6/3/2020   7:56 PM            3420 pom.xml
-a-----         6/3/2020   7:56 PM            1732 README.md
-a-----         6/3/2020   7:56 PM             397 useful_git_command.txt
```

Locate to the target folder

```
PS C:\Users\charl\Desktop\git\demo\APARAPI-acceleration\target> dir

Directory:
C:\Users\charl\Desktop\git\demo\APARAPI-acceleration\target

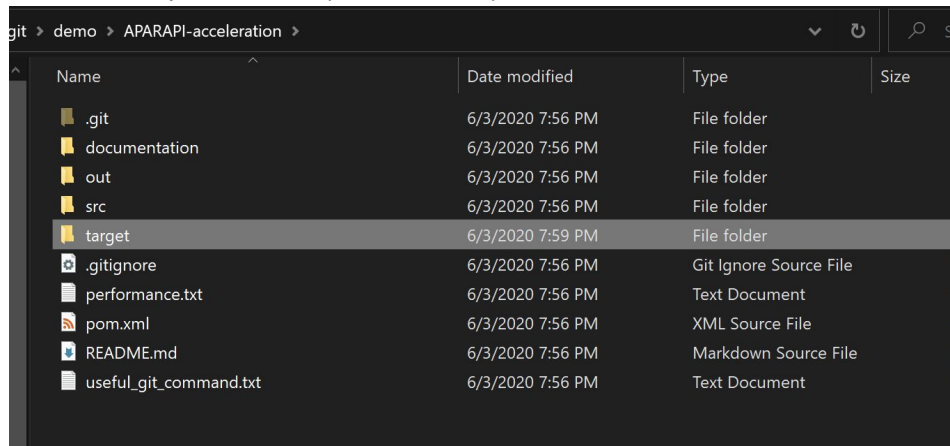
Mode                LastWriteTime         Length Name
----                -
d-----         6/3/2020   7:59 PM                archive-tmp
d-----         6/3/2020   7:59 PM                classes
d-----         6/3/2020   7:59 PM                generated-sources
d-----         6/3/2020   7:59 PM                generated-test-sources
d-----         6/3/2020   7:59 PM                maven-archiver
d-----         6/3/2020   7:59 PM                maven-status
d-----         6/3/2020   7:59 PM                test-classes
-a-----         6/3/2020   8:00 PM      8925969 APARAPI-acceleration-1
                        .0-SNAPSHOT-jar-with-d
                        ependencies.jar
-a-----         6/3/2020   7:59 PM      127622 APARAPI-acceleration-1
                        .0-SNAPSHOT.jar
```

Execute command `java -jar`

APARAPI-acceleration-1.0-SNAPSHOT-jar-with-dependencies.jar

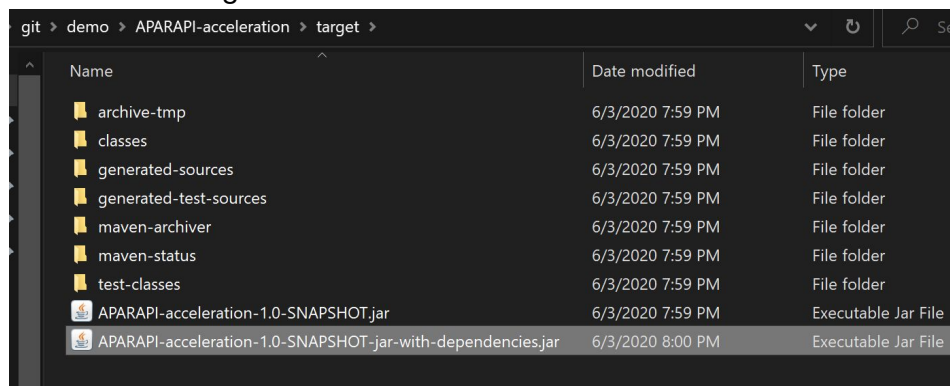
```
PS C:\Users\charl\Desktop\git\demo\APARAPI-acceleration\target> java -jar
.\APARAPI-acceleration-1.0-SNAPSHOT.jar
```

Windows Explorer View (Windows 10)



Name	Date modified	Type	Size
.git	6/3/2020 7:56 PM	File folder	
documentation	6/3/2020 7:56 PM	File folder	
out	6/3/2020 7:56 PM	File folder	
src	6/3/2020 7:56 PM	File folder	
target	6/3/2020 7:59 PM	File folder	
.gitignore	6/3/2020 7:56 PM	Git Ignore Source File	
performance.txt	6/3/2020 7:56 PM	Text Document	
pom.xml	6/3/2020 7:56 PM	XML Source File	
README.md	6/3/2020 7:56 PM	Markdown Source File	
useful_git_command.txt	6/3/2020 7:56 PM	Text Document	

Locate to the target folder



Name	Date modified	Type
archive-tmp	6/3/2020 7:59 PM	File folder
classes	6/3/2020 7:59 PM	File folder
generated-sources	6/3/2020 7:59 PM	File folder
generated-test-sources	6/3/2020 7:59 PM	File folder
maven-archiver	6/3/2020 7:59 PM	File folder
maven-status	6/3/2020 7:59 PM	File folder
test-classes	6/3/2020 7:59 PM	File folder
APARAPI-acceleration-1.0-SNAPSHOT.jar	6/3/2020 7:59 PM	Executable Jar File
APARAPI-acceleration-1.0-SNAPSHOT-jar-with-dependencies.jar	6/3/2020 8:00 PM	Executable Jar File

Double click on it to execute it

