

Documentation by: Amin Shokrzadeh

HR: Masuma Rahim

Answer to assessment question (creatingly.tech)

Assessment Question

Task: Write a simple image processing algorithm in C++ Or Rust that performs a color manipulation on a large image dataset efficiently.

Key evaluation points:

- * Design an algorithm that applies the color manipulation to each pixel of the image dataset in the most efficient way possible.
- * Utilize data structures or mechanisms to efficiently handle the image dataset in memory, considering its size and the necessary operations.
- * Utilize multithreading to process different sections of the image dataset concurrently. Show how parallelism enhances the performance of your algorithm.
- * Optimize your code to minimize any bottlenecks, enhance cache performance, or reduce unnecessary operations. Identify and improve areas of the code that could hinder performance.
- * Utilize a custom data structure or combination of data structures to manage intermediate results efficiently during image processing. Compare its performance to using standard data structures.
- * Implement file I/O operations to read and write image data. Use buffered I/O techniques to enhance the performance of data transfer. Compare the execution time between buffered and unbuffered I/O.

Note: You can read the image from the hardcoded path without the need to visually upload the image. This approach eliminates the need for client/server programming and maintains minimal code.

Notes:

I'm glad to get to know creating team and happily to be a candidate to do this assessment. There was no deadline to finish the assessment and it was mentioned "simple" in title of the task without any certain goal to achieve (in terms of compute time & space complexity) and so I tried to take it simple and finish it as soon as possible. The implementation can be improved to achieve a faster processing algorithm, and I can improve it as needed if there is more time to finish it.

Kindly, there are some notes to mention:

- a) It was asking "Implement file I/O operations to read and write image data. Use buffered I/O techniques to enhance the performance of data transfer. Compare the execution time between buffered and unbuffered I/O."
and so, I implemented my bitmap class to read/write and manipulate bitmap files. I chose bitmap format as it is a popular format for image processing and bmp format is simple and easy to understand. The implemented bitmap class is simple and supports to read/write 24bits and 32bits bitmap images with no compression (normal windows bitmap images).
- b) It was asking "Design an algorithm that applies the color manipulation to each pixel of the image dataset in the most efficient way possible."
And so, I chose to manipulate images by change colorful images to grayscale images as it applies the color manipulate to each pixel of the image. I tried to do it in the most efficient way, but it can be still more efficient.
- c) It was asking "Utilize data structures or mechanisms to efficiently handle the image dataset in memory, considering its size and the necessary operations."
And so, I implement a bitmap buffer (bmpbuffer.h) class to manage bitmap images in memory with some simple functions.
- d) It was asking "Utilize multithreading to process different sections of the image dataset concurrently. Show how parallelism enhances the performance of your algorithm."
And so, I utilized a simple multithreading approach to fetch and process multiple bitmaps simultaneously, and it was so cool and more efficient (in compare to single thread approach). It can be really efficient if use different HDDs to read/write bitmap files.
- e) It was asking "Optimize your code to minimize any bottlenecks, enhance cache performance, or reduce unnecessary operations. Identify and improve areas of the code that could hinder performance."
I tried to optimize my code, but I didn't have much time (I was worry about unknown deadline) to identify and bottlenecks and enhance performance, and I believe there are more improvements to do.
- f) It was asking "Utilize a custom data structure or combination of data structures to manage intermediate results efficiently during image processing. Compare its performance to using standard data structures."

And so, I implement some structures to save bitmap data, pixel, buffer cell, ... but actually I had no idea about “manage intermediate results efficiently during image processing”, as it was a simple RGB to gray image processing algorithm (inside two nested loops) without any specific intermediate data for the next step of processing, and so I’m sorry if I didn’t understand it.

Thank you for studying

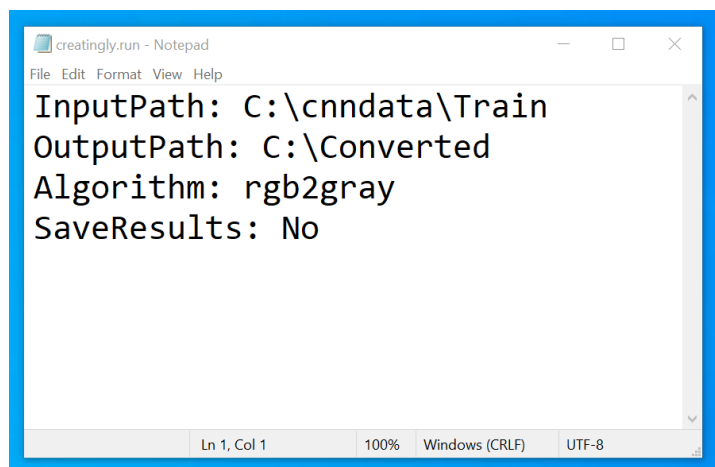
Best regards

Amin Shokrzade

How to Run step by step:

Step1: Creating a run file

- a) Please create an empty file and give it a name (e.g., creatingly.run). There should be four rows inside the file.
 - a. InputPath: Path to directory of input bitmap images
 - b. OutputPath: Path to directory to save result bitmap images
 - c. Algorithm: The image processing algorithm and it can be rgb2gray or rgb2bin
 - d. SaveResults: To determine Save the results or not. It can be Yes or No. it is useful when we want to ignore save file time and calculate read file and processing times only.
 - e. It would be looks like this

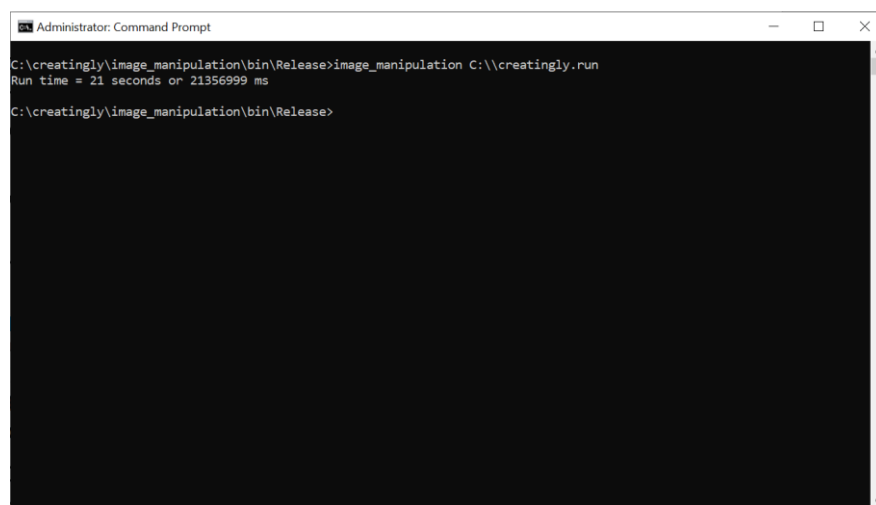


```
creatingly.run - Notepad
File Edit Format View Help
InputPath: C:\cnndata\Train
OutputPath: C:\Converted
Algorithm: rgb2gray
SaveResults: No
Ln 1, Col 1 100% Windows (CRLF) UTF-8
```

Note: keywords are case sensitive!!!

Step2: Execute application

- a) Open command prompt (as administrator recommended)
- b) Enter image_manipulation <run file path> (e.g., image_manipulation C:\\creatingly.run)



```
Administrator: Command Prompt
C:\creatingly\image_manipulation\bin\Release>image_manipulation C:\\creatingly.run
Run time = 21 seconds or 21356999 ms
C:\creatingly\image_manipulation\bin\Release>
```