

Faculty of Applied Sciences  
Bachelor of Science in Computing

**COMP490 Final Year Project  
Progress Report**Academic Year 2022/23

|  |  |
| --- | --- |
| Shader-Based Image Processing Tools | |
|  |  |
| Project number: | Your project number |
| Student ID: | P1908351 |
| Student Name: | Versa Xu |
|  |  |
| Supervisor: | Your supervisor |
| Assessor: | Your assessor |
|  |  |
| Submission Date: | Submission date |

Declaration of Originality

I, Versa Xu, declare that this report and the work reported herein was composed by and originated entirely from me. This report has not been submitted in any form for another degree or diploma at any university or other institute of tertiary education. Information derived from the published and unpublished work of others has been acknowledged in the text and a list of references is given in the bibliography.

[Your signature]

[Date of signature]

Abstract

Text highlight in green are instruction or hints. Text highlight in gray are sample text to demonstrate formatting. The following paragraph is an example.

Sample text sample text Sample text sample text Sample text sample text Sample text sample text. Sample text sample text Sample text sample text, Sample text sample text Sample text sample text.

There are also some placeholder highlight in red. Change them to your own information, e.g. your name and project title.

In any submitted report, you must delete or replace all the colour text.

Here is a placeholder to be finished after all the chapters are well finished.

Table of Contents

[1 Introduction 6](#_Toc119496741)

[1.1 Objectives 7](#_Toc119496742)

[1.2 Risk Assessment 7](#_Toc119496743)

[1.3 Summary 11](#_Toc119496744)

[2 Background and Related Work 12](#_Toc119496745)

[2.1 Digital image processing 12](#_Toc119496746)

[2.2 Shader 14](#_Toc119496747)

[2.2.1 Vertex Shader and Fragment Shader 15](#_Toc119496748)

[2.2.2 GPU vs CPU 16](#_Toc119496749)

[2.2.3 WebGL vs three.js 17](#_Toc119496750)

[2.3 Domain 2 shader/ GPU/ shader language/ workflow/ 3.js/ webgl 17](#_Toc119496751)

[2.4 Related Work 17](#_Toc119496752)

[3 Completed Work (design -> UI, component/ filter: function -> book to reference / ) 19](#_Toc119496753)

[3.1 First Topic 19](#_Toc119496754)

[3.2 Second Topic 19](#_Toc119496755)

[4 On-going and Future Work 20](#_Toc119496756)

[4.1 First Topic 20](#_Toc119496757)

[4.2 Second Topic 20](#_Toc119496758)

[5 Conclusion 21](#_Toc119496759)

[References 22](#_Toc119496760)

Table of Figures

[Figure 1: Probability impact matrix before proposed solution 1](#_Toc454957831)

List of Tables

[Table 1: Table of prioritized risk 1](#_Toc449539554)

# Introduction

Introduction must include the following:

* Background and motivation. Problem. Related works
* Project description
* Objectives
* Main tasks

Refer to the Writing Guide and the Writing Workshop for more detail on the content requirement for each chapter.

As a representation of objective objects, Image is the most used information carrier in human social activities, and the most important means for people to obtain information, express information and transmit information. According to statistics, about 75% of a person's information comes from visual information. Nowadays, the technologies for processing digital images have become more and more advanced. Widely used in the traditional field of agriculture, military, industry, medical science, and so on, digital image processing plays a significant role in dealing with a huge number of pictures containing fragmented information. Particularly, some advanced techniques such as Artificial Intelligence, Autonomous driving, and High precision and high-resolution identification are in urgent need of high-speed, real-time, and precise GPU algorithms to process various kinds of pictures captured by the carried camera. Based on the application and development trends of digital image processing, I think it is feasible to create a multi-functional application, which can improve the availability of raw digital images and provide to the subsequent procedure of research or business transaction. The evolvement of Computers, the discovery of mathematics, and the expansion of applications are three main fields that affect its development and innovation, which decide that the main task is the corporation of GPU computing ability, matrix algorithm, and adaption of electronic equipment.

## Objectives

Write one paragraph to state the aim or goal of the whole project. Then break down the goal into 4-6 SMART objectives.

The project aims to design and generate some graphics processing tools under the functionality of GPU on an interactive online software. The major objectives of this project are listed as follows:

* Study the characteristics of digital images and relevant processes of how to deal with the multidimensional array in Shader Language.
* Design and implement some digital image processing algorithms as shaders running on GPUs and ensure their availability and efficiency at a prominent level.
* Design and implement a user-friendly and interactive online application that supports the predefined digital image processing functions, and allows users to make some changes to their uploaded graphics, including image flick, image twist, image enhancement, and image blur, etc.
* Provide an approach for users to arrange and manage the uploaded images, in an organized and clear interface to browse those original graphics and results.
* The algorithms and the applications can protect the personal information of users and their uploaded images.
* (Optional) Researching matrix algorithms and image security to improve overall performance.

## Risk Assessment

**Risk 1:** **Version management confusion**

Different versions of the same project affect the processes of system design as the overall website runs compatibly with many functions. If each iteration does not update the algorithms or interface components from the previous version, user may not able to correctly figure out the availability and functionality of each function. What’s worse, not updating all functions at the same time in different versions can lead to a chaotic and unmanageable project.

Solution: When the project has many kinds of functions or usage model, we have to ensure that all the things that needs to be updated are listed clearly and updated synchronously. And the project should inform the user what the new version have changed compared to the previous one, in order to avoid the accidents that users’ experience drops because of unwilling to update.

**Risk 2: Users rejects the product due to the unfriendly user interface**

The user interface of the application is not user-friendly, such as poor interface design, which makes users confused and difficult to apply those functions, which may eventually cause users to refuse to use the application.

Solution: In the design phase, follow some widely used design principles, such as The Eight Golden Rules [6]. Also in the product development process, arrange some UI-related tests and find some users with IT background for the first stage of testing and use, try to find out what may be unfriendly to ordinary users and make improvements.

**Risk 3: System do not support the different device**

The digital images processing website is designed to be used in devices with GPU while some people may use it in mobile devices. These devices may not abl [1]e to run shaders on GPU, causing the failure of running algorithms. In the meanwhile, software and hardware incompatibilities should be considered. In many situations such as power saving mode and non-performance mode, the user may not be able to perceive the benefits of using GPU to process digital images.

Solution: Take widely testing before implementation. Also give the user some hints of their current devices before really applying the project environment.

**Risk 4: User data breach**

The user upload photo and some related information are just shown on the web page, so they should not be leakage for other people know. But the hacker may use illegal ways to get these user data or delete the data.

Solution: Backup images prevent a data breach, and regular checking of the website detects the aberrant download action. Enhance the HTTP security.

**Risk 5: Computer failure or development environment problems**

If the computer fails in the development process, the whole project may be stopped. This will seriously affect the progress of the project.

Solution: Make good use of online code repositories such as Github and make regular backups to ensure that the development progress will not be affected even if the computer fails.

**Table of Priority Risk**

|  |  |
| --- | --- |
| Priority | Risk identifier and Description |
| 1 | **Risk 5: Computer failure or development environment problems** |
| 2 | **Risk 4: User data breach** |
| 3 | **Risk 1: Version management confusion** |
| 4 | **Risk 2: User rejects the product due to the unfriendly user interface** |
| 5 | **Risk 3: System do not support the different device** |

Notes: Priority 1 is the highest risk

**Table 1: Table of priority risk**

As shown in table 2, there are mainly five related risks. The highest risk is computer failure or development environment problems, which may make the whole project use the wrong programming language or developing framework. It is necessary to apply precise developing tools and environment from the initial process and make sure all the codes are carefully saved after executing. The second biggest risk is user data breach. If the website has no good security, some individual or important images which include private information will be stolen by hacker. Therefore, users will no longer trust the website and feel worried. What’s more, confusing versions may also provide the user with poor using experience, since without a clear version description or updating information, it is hard to recognize the new functions and related changes in the old ones. It is also possible that users might reject the product because of not well-designed interfaces. This can really impact the impression of the website for the first time. Finally, we consider that the system may not be suitable for all the devices as only some PC or laptop have a GPU.

**Probability Impact Matrix (****Initially)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Probability | High |  | Risk 5 |  |
| Medium |  | Risk 4, Risk 2 | Risk 1 |
| Low |  | Risk 3 |  |
|  | | Low | Medium | High |
| Impact | | |

**Figure 1: Probability impact matrix initially**

Figure 1 shows the initial risk probability impact, this project occurs in risk5 which has a high probability, risk 4 and risk 2 have medium probability, and risk 3 has a low probability. As see the impact way, risk2,3,4,5 also has medium impact, risk 1 is high impact.

**Probability Impact Matrix (After Applying Solution)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Probability | High |  |  |  |
| Medium |  |  |  |
| Low | Risk1, Risk 2, Risk 3 | Risk 4, Risk 5 |  |
|  | | Low | Medium | High |
| Impact | | |

**Figure 2: Probability impact matrix after applying solution**

Figure 2 shows that the risk probability risk after the applying solution, risk 1, risk 2, risk 3 just have a low probability risk and impact. Risk 4, risk 5 also have a medium impact, so it needs to focus on it to prevent this risk occurs.

Table 1: Table of prioritized risk

|  |  |
| --- | --- |
| Priority | Risk Identifier and Description |
| 1 | Risk 1: short description |
| 2 | Risk 2: short description |
| 3 | Risk 3: … |
| 4 | Risk 4: .. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Probability** | High |  |  | Risk 1 |
| Medium |  | Risk 3 | Risk 2 |
| Low |  |  |  |
|  |  | Low | Medium | High |
|  |  | **Impact** | | |

Figure 1: Probability impact matrix before proposed solution

## Summary

The summary should be finished like this: This report is organized as follows: Chapter 2 introduces the background of our work. Chapter 3 presents our design approach. Chapter 4 shows the implementation details….

# Background and Related Work

In this chapter, you provide background information for readers to help them understand your project. There may be more than one sections on background domain.

This chapter also provides detail about related work. In chapter 1, you should have mentioned some related works and explain how your project is related to them when you discuss relevancy. If you want to provide detail of related works, include them in this chapter.

Call this chapter “Background” if all related works have been described in Chapter 1.

在当前社会发展中，GPU图像处理在很多基础技术和高级技术领域至关重要：(此处列举xxxx based on digital image processing) image processing

图像处理的定义、发展历程、关键技术

CPU vs GPU

算法， shader(可以写多一点，gpu,画图)

应用程序

## Digital image processing

An image may be defined as a two-dimensional function. f(x, y), where x and y are spatial coordinates, and the amplitude of f at any pair of coordinates (x, y) is called the intensity or gray level of the image at that point. When x, y, and the intensity values of f are all finite, discrete quantities, we call the image a digital image. The field of digital image processing refers to processing digital images by means of a digital computer [2]. Digital image processing is the use of a digital computer to process digital images through an algorithm. With the development of science and technology and the improvement of computer performance, in different fields, it is necessary to use image processing technology to extract the effective information in the image, repair, restore and noise processing of the image, to improve the image quality and analyse and predict related problems.

Digital image processing originated in the 1920s, when a photograph was sent via an undersea cable from London, England, to New York, United States, using digital compression technology. As a discipline, digital image processing was formed in the early 1960s. The purpose of early image processing was to improve the quality of images, and it took people as objects to improve people's visual effects. The first big success came from JPL, which used image processing on thousands of images of the moon sent back by the space probe Prowler 7 in 1964 and successfully mapped the moon's surface by computer. This was followed by more sophisticated image processing on nearly 100,000 images sent back by the spacecraft. The topographic map, colour map, and panoramic Mosaic map of the moon were obtained, which laid a solid foundation for the pioneering work of humans landing on the moon. In future space technology, such as the exploration of Mars, Saturn, and other planets, digital image processing technology has played a huge role. In 1972, using image reconstruction, according to the projection of the human head section, through computer processing to reconstruct the section image for skull diagnosis X-ray computerized tomography device, namely CT (Computer Tomograph), was invented by British EMI engineer Hounsfield. In 1975, EMI Company successfully developed a CT device for the whole body and obtained distinct and clear sectional images of various parts of the human body. In 1979, the damage-free diagnostic technique won a Nobel Prize. With the in-depth development of image processing technology, since the mid-1970s, with the rapid growth of computer technology and artificial intelligence, thinking science research, digital image processing to a higher and deeper level of development.

This field includes many critical and fundamental techniques, such as image quality enhancement, image analysis, image reconstruction, and image compression, etc.

* Image quality enhancement: Image quality improvement is to remove the distortion and noise information on the image as far as possible, to make the image clearer, to accurately interpret and interpret the image information visually. Sharpening, smoothing, recovering, revising.
* Image analysis: Image analysis generally refers to the use of mathematical models and image processing technology to analyse the underlying features and superstructure, to extract information with a certain intelligence. The purpose of image analysis is to extract useful information from the image, the commonly used techniques include edge and line detection, image region segmentation, shape feature extraction and measurement, image texture analysis, image matching, fusion, etc.
* Image reconstruction: Image reconstruction is a mature and practical image processing technology, which is to obtain information on material distribution in the scene according to the projection data of the scene. Widely used in the medical field, it mainly includes the 3D reconstruction of the projected image in CT, and the technology used to measure the left and right view images and generate a stereoscopic image.
* Image compression: Image data compression is aimed at the image data generated by the digital information is very large, especially the colour dynamic image data amount is surprisingly large, in order to transfer and preview these images, it is necessary to reduce the image storage capacity, in order to save image transmission, processing time and reduce the occupied memory capacity. Compression can be achieved without losing truth or with distortion allowed. Commonly used static image lossless compression and lossless compression technology, such as WinZip, WinRAR, and a variety of image format conversion; There is also dynamic image compression processing technology, such as MPEG, H.264, AVS, network streaming media technology, and so on. Coding is the most important method in compression technology, and it is the earliest and relatively mature technology in image processing technology.

实现细节：矩阵处理、人工智能图像处理等，对比

## Shader

Shaders are also a set of instructions, but the instructions are executed all at once for every single pixel on the screen. That means the code you write must behave differently depending on the position of the pixel on the screen. Like a type press, the program will work as a function that receives a position and returns a colour, and when it's compiled it will run extraordinarily fast. GPU shaders are generic computer programs written to be executed on a graphics processing unit (GPU). Shaders were originally programs running on hardware parts of GPUs which were specifically designed to do actual shading in 3D graphics. Modern "shaders" are executed on more general-purpose CUDA cores on [Nvidia](https://linuxreviews.org/Nvidia) GPUs and GCN processors on [AMD](https://linuxreviews.org/Advanced_Micro_Devices) GPUs.

The AMD documentation for their Vega 7nm GPU products launched in 2019 describe "shaders" as:

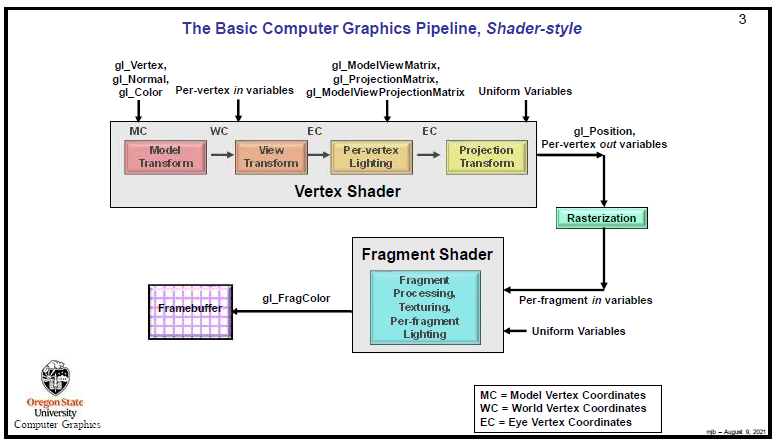
*"Compute kernels (shaders) are generic programs that can run on the GCN processor, taking data from memory, processing it, and writing results back to memory. Compute kernels are created by a dispatch, which causes the GCN processors to run the kernel over all the work items in a 1D, 2D, or 3D grid of data. The GCN processor walks through this grid and generates wavefronts, which then run the compute kernel. Each work-item is initialized with its unique address (index) within the grid. Based on this index, the work-item computes the address of the "Vega" 7nm Instruction Set Architecture data it is required to work on and what to do with the results."*

[*"Vega" 7nm Instruction Set Architecture Reference Guide*](https://gpuopen.com/wp-content/uploads/2019/11/Vega_7nm_Shader_ISA_26November2019.pdf) *published November 26th, 2019*[[1]](https://linuxreviews.org/GPU_shaders#cite_note-1)

It is of note that AMD refers to "shaders" as "**Compute kernels (shaders)**".

### Vertex Shader and Fragment Shader

* **Vertex Shader**: it is the programmable Shader stage in the rendering pipeline that handles the processing of individual vertices. In graphics the vertex programs are usually used to perform the calculations, that can be interpolated linearly over a vertex, in order to save computations in the fragment program. A vertex shader receives a single vertex to the output vertex stream and generates a single vertex to the output vertex stream. There must be a 1:1 mapping from input vertices to output vertices [3].
* **Fragment Shader**: it is the Shader stage that will process a Fragment generated by the Rasterization into a set of colors and a single depth value. Fragment shader takes care of how the pixels between the vertices look. They are interpolated between the defined vertices following specific rules [4].



**Figure 3: The workflow of a shader**

Figure 3 describes how a shader process the tasks on the GPU through OpenGL [5]. The roles of vertex Shader and Fragment Shader in the OpenGL pipeline runs differently compared with the fixed-functions. Vertex Shader will replace the sequential steps of model transform, view transform, per-vertex lighting, and projection transformation with some built-in variables. The *gl\_Position* and out variable created from vertex shader undergo the rasterization process to create a fragment for every pixel.  Fragment Shader is used in variable, uniform variable, and local variables to perform fragment processing, texturing, and per-fragment lighting. *gl\_FragColor* produced from fragment Shader then get into the framebuffer.

### GPU vs CPU

Most shaders are coded for and run on a graphics processing unit (GPU). Shader languages are used to program the GPU’s rendering pipeline, which has mostly superseded the fixed-function pipeline of the past that only allowed for common geometry transforming and pixel-shading functions. Using a shader can allow the program to take advantage of the processing power of the GPU instead of relying on only the system CPU, which performs well in latency because it allows parallel pixel processing. We know that a CPU is like the computer’s brain and GPU serves to be its soul, but the processors also have differentiating factors in their functionalities. There are several significant criteria that can define whether the image processing hardware is:

* **Latency:** The reduction in latency is possible with GPU architecture because it allows parallel pixel processing.Because the CPU has parallel processing but on the levels of tiles, frames, and image lines, the latency tends to remain.
* **Performance:** The ideal level of performance can either be achieved by optimizing the code of the software or by increasing hardware resources, which means the number of processors. Therefore, GPU stands out to be better than CPU in regards of price-to-performance ratio. However, the real potential pf GPU is explored only with multilevel algorithm or parallelization.
* **Quality of image processing:** Even though multiple algorithms can be used to perform the same image processing, it will differ in the quality of result and resource intensity.

|  |  |  |
| --- | --- | --- |
|  | **CPU** | **GPU** |
| **Processing Type** | Serial instruction processing | Parallel instruction processing |
| **Thread Numbers** | An individual thread executes an instruction solely | Group of thread execute the instructions |
| **Core** | One but strong | Thousands but weak |
| **Implementation of Thread** | Out-of-order execution method | Genuine thread rotation |
| **Latency & Throughput** | Low latency, low throughput | High latency, High throughput |
| **Utility** | For wide range processing | For rendering 3D graphics |

**Table 2: Comparisom between CPU and GPU**

The above Table.2 shows many suitability and feasibility of using GPU to do digital image processing. More importantly, almost every modern GPU has shared memory, making it way better and exponentially faster than the CPU’s cache. It works best with algorithms that have a high degree of locality. What’s more, a GPU can considerably reduce the load on a subsystem by modifying the number of registers. The power of parallel processing makes GPU faster than CPU when process huge amounts of high-quality graphics, as it owns better memory and processing power bandwidth [6].

### WebGL vs three.js

fhf

## Domain 2 shader/ GPU/ shader language/ workflow/ 3.js/ webgl

## Related Work

Give a brief description of related works. May be omitted if enough detail of related works are already covered in Chapter 1.

Digital image processing 在论文中的应用领域。

The wild applications of digital image processing (DIP) can be discovered in many fields, and this technique has made essential contributions in the progress. In digital mammography, DIP has a significant affected by the type of image processing. The purpose of this work is to perform a retrospective observer study to investigate the impact of different image-processing algorithms on the detection of cancers in digital mammography [6]. Images of calcification clusters were inserted into breast images that had originally showed normal findings. Through this way, the system improves the performance of detecting a targeted cancer, by contrasting the Reader-Averaged Figure of Merit (FOM) between standard, film-screen, and low contrast group. Xin Tand and Wei Chen (2016), developed a numerical analysis method based on the combination of discrete element method and digital image processing. They use it to reproduce the mineral composition of the rock and used the processed image as input to set up a corresponding numerical discrete element model [7]. The technique applied in this project is indispensable because many original images with a few noises will affect the critical information detection, after that, the images become quite clear and the striations and eagles of the rock can be identified easier. Similarly, Nafaa Nacereddine and M. Tridi, using the method of dynamic stretching and Local contrast enhancement, successfully changed the brightness of the images, by mapping the pixel value to the corresponding higher value evenly at the range of 0~255 [8]. Others like Kenji Omasa and Morio onoe, who developed a digital image technique for exactly measuring the degree of stomatal opening – detect the ratio of the width to the maximum length of a stomatal pore, and the pore area. [9] All these researches consider the digital image processing as the initial part, before observing and evaluating data, which is usually called pre-processing.

Image processing app 的工作流程 + shader 的工作流程放在design中 。

# Completed Work (design -> UI功能模块框图,显示的东西得描述，render, error message等, component/ filter: function -> book to reference / )

The first paragraph is the introduction paragraph. This paragraph usually gives a brief overview of each section in the chapter. The logic flow behinds the section arrangement should also be described.

This chapter describes ‘Completed works’. These works typically include system analysis, data modelling, system architecture, experiment design, etc. However, this chapter should only include original, creative works. Text that mostly describes others’ works should be moved to Chap 1 related works or Chap 2 background.

Also highlight difficulties encountered, alternatives evaluated and solutions adopted.

Content of this chapter will be distributed to Chap 3, 4 and 5 of the Final Report.

总分或分总

Loader -> process shader -> output

Decision -> 列表， 选择

## First Topic

Sample text sample text Sample text sample text Sample text sample text Sample text sample text. Sample text sample text Sample text sample text, Sample text sample text Sample text sample text.

## Second Topic

Sample text sample text Sample text sample text Sample text sample text Sample text sample text. Sample text sample text Sample text sample text, Sample text sample text Sample text sample text.

# On-going and Future Work

Write an introduction paragraph to delineate the content and logic flow of this chapter.

* Describe partially done works
* Include a Gantt chart as evidence of effective project planning for the 2nd semester
* Show Clear idea of what to do to complete the project

## First Topic

Sample text sample text Sample text sample text Sample text sample text Sample text sample text. Sample text sample text Sample text sample text, Sample text sample text Sample text sample text.

## Second Topic

Sample text sample text Sample text sample text Sample text sample text Sample text sample text. Sample text sample text Sample text sample text, Sample text sample text Sample text sample text.

# Conclusion

Reflect on the progress of the project. Can use first person pronoun to write.

Content may be moved to the Reflection appendix in the Final Report.

Sample text sample text Sample text sample text Sample text sample text Sample text sample text. Sample text sample text Sample text sample text, Sample text sample text Sample text sample text.

# References

|  |  |
| --- | --- |
| [1] | Wei Wang, M. N.s. Swamy, M. Omair Ahmad, “RNS application for digital image processing,” in *4th IEEE International Workshop on System-on-Chip for Real-Time Application(IWSOC'04)*, 2004. |
| [2] | Rafael C. Gonzalez, Richard E. Woods, Digital Image Processing (Third Edition), Prentice Hall, 2007. |
| [3] | “OpenGL,” 10 Nov 2017. [Online]. Available: https://www.khronos.org/opengl/wiki/Vertex\_Shader. |
| [4] | P. D. E. Jensen, Wrokflow Improvements for Real-time Shader Development, Kongens Lyngby: Technical University of Denmark, 2006. |
| [5] | A. Vu, “blogs.oregonstate.edu,” 20 Oct 2021. [Online]. Available: https://blogs.oregonstate.edu/learnfromscratch/2021/10/20/vertex-shader-and-fragment-shader/. |
| [6] | N. Foster, “GPU vs. CPU for Image Processing: Which One is Better?,” Ace Cloud Hosting, 6 10 2022. [Online]. Available: https://www.acecloudhosting.com/blog/gpu-vs-cpu-for-image-processing/. |
| [7] | Lucy Warren, Rosalind Given Wilson, Matthew Wallis, Mark D. Halling-Brown, Julie Cooke, “The Effect of Image Processing on the Detection of Cancers in Digital Mammography,” *American Journal of Roentgenology,* vol. III, pp. 387-393, 2014. |
| [8] | Xin Tan, Wei Chen, “Numerical Simulation of Heterogeneous Rock Using Discrete Element Model Based on Digital Image Processing,” *Rock Mechanics and Rock Engineering,* 2016 June. |
| [9] | Nafaa Nacereddine, M. Tridi, “Weld defect detection in industrial radiography based digital image processing,” 2005. |
| [10] | Kenji Omasa, Morio Onoe, “Measurement of Stomatal Aperture by Digital image processing,” *Plant and Cell Physiology ,* pp. 1379-1388, 1984. |
| [11] | S. Aydemir, “Qualification of soil features using digital image processing(DIP),” *Geoderma,* 2004 March. |
| [12] | “Mdn Web Docs,” Mozilla Corporation, 26 Aug 2022. [Online]. Available: https://developer.mozilla.org/en-US/docs/Games/Techniques/3D\_on\_the\_web/GLSL\_Shaders. |