**FISCA-PART-2 MINI PROJECT**

**Image Processing Project using CUDA:**

**Instructions for Students:**

**Project group size should be 4 or less than 4. (preferably 4) [from your section only]**

**Submission of the project: first / second week of April.**

**Each member in a group is responsible for some part of the project having clear objective.**

**Project will be evaluated for 5marks as (FISAC-2 COMPONENT)**

Introduction:

In this project, you will explore image processing techniques using CUDA, a parallel computing platform and programming model developed by NVIDIA. CUDA allows you to harness the power of NVIDIA GPUs for parallel computing tasks, which can significantly accelerate image processing algorithms compared to traditional CPU-based implementations.

Project Objective:

The objective of this project is to implement various image processing operations, such as filtering, convolution, edge detection, and morphological operations, using CUDA. You will compare the performance of CUDA-accelerated algorithms with their CPU counterparts to demonstrate the benefits of GPU parallelism in image processing tasks.

Instructions:

Understanding CUDA:

Familiarize yourself with CUDA programming concepts, including threads, blocks, grids, shared memory, and thread synchronization.

Learn about CUDA memory hierarchy and memory management functions such as cudaMalloc, cudaMemcpy, and cudaFree.

Selecting Image Processing Algorithms:

Choose a set of image processing algorithms to implement using CUDA. Common algorithms include:

Convolution (e.g., Gaussian blur, edge detection) Filtering (e.g., median filtering, mean filtering)

Morphological operations (e.g., dilation, erosion)

Ensure that the chosen algorithms are suitable for parallelization and can benefit from GPU acceleration.

Implementing CUDA Kernels:

Design CUDA kernels to perform the selected image processing operations. Each kernel should handle a specific part of the algorithm and execute in parallel across multiple threads.

Optimize kernel performance by minimizing memory access latency, maximizing memory coalescing, and exploiting thread-level parallelism.

Memory Management:

Allocate device memory for input and output images using cudaMalloc.

Transfer image data between host (CPU) and device (GPU) memory using cudaMemcpy.

Free allocated device memory using cudaFree when no longer needed.

Kernel Launch Configuration:

Determine the optimal kernel launch configuration (number of threads per block, number of blocks per grid) based on the image size and hardware capabilities.

Experiment with different configurations to achieve the best performance.

Integration and Testing:

Integrate CUDA-accelerated image processing algorithms into a complete application.

Test the correctness and performance of CUDA implementations by comparing results with CPU-based implementations.

Validate the speedup achieved by CUDA acceleration compared to CPU execution.

Performance Analysis:

Measure the execution time of CUDA-accelerated algorithms and CPU-based algorithms for various input image sizes.

Analyze the speedup and efficiency gained by using CUDA for image processing tasks.

Identify any bottlenecks or areas for further optimization.

Documentation and Presentation:

Document your implementation details, including CUDA kernel code, memory management strategies, and performance analysis results.

Prepare a presentation summarizing your project, including objectives, methods, results, and conclusions.

Present your findings to the class, highlighting the advantages of using CUDA for image processing.

Conclusion:

By completing this project, you will gain hands-on experience in parallel programming with CUDA and understand the benefits of GPU acceleration for image processing applications. This project will enhance your skills in algorithm optimization, performance analysis, and parallel computing techniques

**Note: Projects will be scanned for plagiarism. Therefore, students are expected to come up with unique projects.**

**Deadlines:**

1. Topic selection and forming the group (9/3/2024) [A MS-FORM will be shared in which you have to enter your group details and project title, only one member of the group will enter )
2. Submission of the report (Documentation of the project) : 04/04/2024
3. Project Presentation : Second week of April