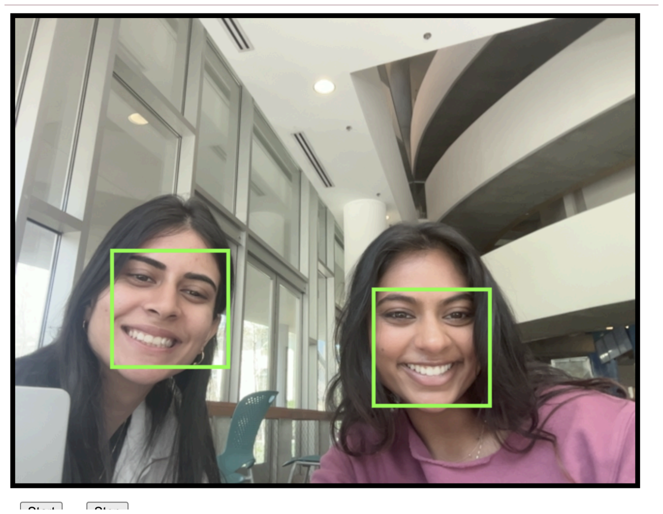
Parallel Face Detection

Ria Manathkar and Gaurika Sawhney
https://parallel-s24.github.io/final-project/

1. Summary of Work So Far

So far, we have a working sequential version of a facial detection system for live video feed. We have been able to test this version, and have some example screenshots below. There is live webcam feed and our program tries to detect faces in frame in real time. The faces detected will be shown with a yellow box outline. We implemented this using a cascade classifier available through OpenCV package, and plan to parallelize the Viola Jones algorithm using CUDA over the next week.



2. Updated Schedule

Make sure your project schedule on your main project page is up to date with work completed so far, and well as with a revised plan of work for the coming weeks. As by this time you should have a good understanding of what is required to complete your project, I want to see a very detailed schedule for the coming weeks. I suggest breaking time down into half-week increments. Each increment should have at least one task, and for each task put a person's name on it.

Week	Item
March 31- April 6	● Set up CUDA environment ● Implement and test the Viola-Jones algorithm sequentially.
April 7 - April 13 [CARNIVAL]	● Research optimization strategies for parallel execution ● Prepare detailed plans for Goals 2.1, 2.2, and 3.
April 14 - April 16	● Implement parallel window processing

	● Explore parallelizing cascade classifier ● Project Milestone Report (Due April 16th)
April 17 - April 20	● Implement parallel window processing (Gaurika) ● Finish implementing CUDA version of facial detector (Ria)
April 21 - April 24	● Implement parallelizing cascade classifier (Gaurika) ● Debug and test CUDA parallel implementation (Ria)
April 25 - April 28	● Debug and test CUDA parallel implementation (Ria) ● Brainstorm cache and memory optimization strategies (Gaurika)
April 28 - May 3	● Optimize cache efficiency (Gaurika) ● Implement memory optimization strategies (Ria) ● Conduct comprehensive testing and performance evaluations (Gaurika & Ria)
May 3 - May 5	● Final Project Report (Due May 5th) ● Project Poster Session (Due May 6th)

3. Goal Reflection

Our goals and deliverables remain the same, though we are a bit behind as we don't have a working version in CUDA yet. We are confident we will produce all the deliverables we initially outlined. However, it may be difficult to achieve our "nice to have" goals, but we plan on working diligently over the next few weeks and if we don't run into any large or time-consuming issues it is still possible for us to achieve.

Plan to Achieve

Goal	Description	Justification
1	Successfully implementing the Viola-Jones algorithm in CUDA.	As a foundational building block for our project, we need to ensure we have a working sequential algorithm that we can use to measure the speedup of following parallel versions we create.
2.1	Assigning window processing (of all sizes) to CUDA blocks so these can all be processed in parallel	By determining all windows with various window sizes at the beginning, we will be able to distribute the work for each window better in terms of CUDA blocks, making the algorithm more efficient
2.2	Parallelize the steps the cascade classifier takes for each integral image	This may be challenging since the cascading classifier is inherently sequential due to layered dependencies. If we are able to parallelize this step somehow, by breaking apart some functions or finding a way around the dependencies, this could help speed up immensely.

3	Improving the amount of memory accesses across the algorithm	As stated above, this face detection algorithm involves many accesses to the input and integral image so overall latency could be greatly impacted by the frequency of memory accesses.
4	Run experiments on both implementations with different problem sizes and input images	We want to determine our performance limitations and the tradeoffs between our two potential methods of parallelization.

Hope to Achieve

Goal	Description	Justification
5	Improving cache efficiency and reducing cache misses despite the distributed work amongst processors	Because each processor will be processing windows across different parts of the image, cache coherency will be a big challenge. We hope to make some optimizations for this, by better distributing work across processors.

4. Poster Session Plan

We plan on making a large poster with several graphs that help us illustrate the performance benefits of our different methods. This means comparing our two methods of parallelizing across window sizes as well as the steps of the cascade classifier for each integral image. We then will demo the facial detection with real-time video input and allow onlookers to manipulate different parameters to see the change in output. Throughout this demo, we will discuss our project journey including significant insights and challenges we ran into.

5. Concerns

One of our primary challenges currently is deploying online trained models for local execution on our computers. We are encountering significant environment setup issues, primarily with certain libraries not being recognized. Beyond that, our largest challenge will be figuring out how to implement our two parallelization strategies.