Portions of project you were responsible for and you delivered on:

1. Jerry
    a. Set up python-c integration, Makefile, modularization, framework for correctness testing nms functions,
    b. Maintained code modularity and consistency
    c. Wrote OMP and SIMD implementations,
    d. Experimented with alternate algorithms/approaches
    e. Performed scaling analysis
    f. Created graphics and graphs for presentation
    g. Developed parallelism strategies for implementation
2. Simon
    a. Setup initial dataset using yolo for testing generic nms code, and wrote the code for reading in datasets
    b. Setup Bichen's large dataset to feed into our existing code and wrote import scripts
    c. Wrote benchmarking code for the large datasets and formatted for easy analysis of our different implementations
    d. Wrote unordered serial c algorithm
    e. Helped debug GPU
    f. Wrote alternative kernel for gpu for splitting work for efficiently
    g. Helped with accurate GPU benchmarking code
    h. Wrote code for computing the Mflops/s
    i. Did the roofline analysis
3. Seth
    a. Configured GPU environment to use Titan X on a remote server
    b. Experimented with different GPUs (hive machine, K40s, Titan X, etc.)
    c. Set up GPU kernel and implemented NMS algorithm on the kernel
    d. Benchmarked GPU performance
    e. Wrote powerpoint presentation + drew graphics
    f. Wrote nmsModule code for the GPU part
4. Tushar
    a. Coordinate with Project Leader
    b. intermediate between project leader and team
    c. GPU skeleton code
    d. w/ some nmsModule skeleton code
    e. a lot of GitHub PR handling
    f. Worked on GPU issues w. team regarding overhead from data transfer
    g. Looked at problems with GPU data transfer overhead alone versus CPU computation speed
5. Lichang
    a. Due to tremendous course pressure from my other five technicals, I did not contribute much to the implementation part of the project. I did follow up with the code base other teammates contributed over time since I have previously done several research projects in computer vision and have used serialized version of NMS from time to time.
6. Vivian
    a. Translated base code from Python to C
    b. Helped set up nmsModule skeleton
    c. Wrote OMP and SIMD implementations
    d. Wrote/formatted tests.py, cleaned up/formatted code
    e. Helped debug GPU + ran tests

f. Set up/wrote powerpoint presentation + drew the graphics

Portions of project that the team member was responsible for, but failed to deliver:

1. Jerry
    a. Wrote early benchmarking code, but Simon's was better
    b. Optimize a communicating multithreaded implementation (master worker?), ran out of time for this
2. Simon
    a. Integrate accuracy tests into the benchmarking
    b. Create visualization of outputs
3. Seth
    a. GPU implementation could have been better
    b. Tried to do multi-GPU NMS, but did not have enough time to fully implement
4. Tushar
    a. GPU skeleton was bit buggy
    b. Tried going from C to CPP but had difficulties with makefile, and C was better
5. Lichang
    a. Did not keep up with the project schedule (including absence from project meeting).
    b. I planned to implement a SIMD + OMP parallel version with improvements over my previous research project but failed to deliver in time.
6. Vivian
    a. Translated CPP module to C, but we ended up not needing it
    b. Wrote a size-4 SIMD, but size-8 was faster

As a team member did they show: reliability, flexibility, accountability, friendliness? As a project leader did they show: honesty, competence, leadership? Finally, comment on their ability to communicate clearly (or not). Also, discuss communication skills.

1. Jerry- I tried to ensure that the code base was easy to work on/add to by continuously maintaining modularity and consistency across different work items. I regularly committed changes and was quick to review my teammate's work. I was an active participant at all group meetings, and helped direct our strategy in parallelizing NMS. Collaborating with my team members was easy.
2. Simon: I set up times to meet as a group and communicated frequently on messenger giving updates of what I was working on and to check in with other people. I always tried to help out with debugging if someone was having a problem.
3. Seth: I actively communicated with other team members to discuss on how to proceed on the project. We would have offline group meetings to which I would participate. I would also state clearly of what I can deliver and try my best to accommodate any feedbacks. Overall, my team members were very collaborative and friendly that it was easy to communicate with them.
4. Tushar: Leadership - helped organize and initiate project,and communicate with project leader; being the conduit between the project lead and team required clear communication skills. Like other team members, also showed other stated traits (reliability, flexibility, competence).
5. Lichang
    a. I did plan to be reliable to contribute to the project since I have previous experiences in the chosen topic. However I did not expect my other courses to be significantly heavier in the middle of the semester and failed to keep up with the project. There is no excuse for this and I'm really sorry for my conflicting schedules. I did have tremendous respect for other five teammates since they have done great job within such a short time. I did try to parallelize NMS before taking this

course once but only achieved roughly 5x speedup. I hold myself accountable for my lack of contribution in this group project.

6. Vivian

SOFTWARE SKILL AREAS

1. Correct coding
    a. Understanding invariants
    b. Test before you code
    c. Applying unit testing, integration testing, system testing
    d. Refactoring
    e. Daily system and Regression testing
2. Computationally efficient coding
    a. Efficient coding techniques
    b. Good algorithms
    c. Problem formulation
    d. Profiling of code
    e. Refactoring
    f. Use of parallelism
3. High-productivity coding
    a. Reuse of software
    b. Do the simplest thing that could possibly work
    c. Ability to quickly use program generators, third party software, or other programming tools
4. Facility with Patterns
    a. Overall architecting ability
    b. 'Structural and computational patterns
    c. Parallel Algorithm Patterns
    d. Implementation Patterns
    e. Target HW Patterns

1) AREA OF STRENGTH/: Describe any areas of strength.

1. Jerry
    a. 1b. Wrote first tests before even starting new implementations
    b. 1d. Consistently refactored my and teammates' code to maintain consistency and modularity
    c. 2b. Good algorithms: Brainstormed and tested new NMS algorithms

   d. 2f. Parallelism: ^ Same as above
   e. 3a. Reuse: Maintained modularity to ensure code reuse
   f. 3b. Simplest thing that could possibly work: Wrote quick, dirty benchmarking code for sanity check before Simon finished the final benchmarking suite
   g. 4c. Parallel Patterns: Implemented various parallel patterns in testing different algorithms
2. Simon
   a. **1c** Correct Coding: Applying unit testing, integration testing, and system testing by writing scripts for testing the correct implementation of the code **2d** profiling of code: wrote benchmarking code for different algorithms **3a** Reuse of software: benchmarking code was easily reusable for different implementations **4a** Overall architecting abilities Test framework was efficient and easy to use
3. Seth
   a. 1.a (Identified what should change and what should remain unchanged in the GPU kernel code)
   b. 2.e (Refactored the code used in homework to efficiently implement kernel)
   c. 3.a (The kernel code could be used for different environments with various GPUs)
   d. 3.b (Tried naïve implementation of NMS first)
   e. 3.c (Use of OpenCL library inside Nvidia CUDA library)
   f. 4.e (Use of multiple GPU types such as Titan X, K40s)
4. Tushar
   a. **2A** - Efficient coding with GPU skeleton; **2C, 2F** - Problem formulation and parallelization when discussing GPU parallelization and overhead; **3A, 3B** - Reuse of software with GPU skeleton; **4A** - when discussing parallelization with GPU
5. Lichang
   a. 1c - achieve modularity or API-driven testing by writing up ad-hoc unit testing programs
   b. 3a - reuse of software by utilizing previous research code to benchmark performances
   c. 4b,c,d,e - have a solid understanding about OPL
6. Vivian

2) AREAS FOR IMPROVEMENT: Describe any areas of weakness not mentioned by team member in their self-assessment..

1. Jerry
   a. 1e. Did not really perform daily testing
   b. 1d. Profiling: At one point, failed to identify memory transfer bottlenecks
2. Simon
   a. **3c** Had difficulty getting third party script working to analyze accuracy
3. Seth
   a. 1.c (I would do test-driven development where I first write unit tests and then start implementing)
   b. 2.d (I would have to run many ablations when benchmarking time / space complexity of a program)
   c. 4.c (I would have to read more about parallel algorithm implementations that are used in the industry)
4.
5. Tushar
   a. Could write more tests code
6. Lichang
   a. Should spend more time in the implementation part

b. should spend more time communicating with other five teammates and project advisor about my concerns in the project development

7. Vivian