Teaching Accelerated Computing with Hands-on Experience

Işıl Öz
Computer Engineering Department
Izmir Institute of Technology
Izmir, Turkey
isiloz@iyte.edu.tr

Chelsea Cropper
Developer Programs
NVIDIA
New York, USA
cpettigrew@nvidia.com

***Abstract*—Heterogeneous computing systems maintain high-performance executions with parallel hardware resources. Graphics Processing Units (GPUs) with many parallel efficient cores and high-bandwidth memory structures enable accelerated computing for high-performance, deep learning, and embedded programs from diverse domains. The expertise in GPU programming requires a significant effort to utilize parallel computational units efficiently. Teaching programming for heterogeneous systems also becomes difficult due to dedicated hardware requirements and up-to-date course materials. In this paper, we present our teaching experience in an undergraduate parallel programming course, where we adopt NVIDIA Deep Learning Institute workshop and teaching kit contents and GPU devices at different scales to expose students to a set of hardware platforms with hands-on coding experience.**

***Index Terms*—accelerated computing, GPU programming, NVIDIA Deep Learning Institute**

I. Introduction

Graphics Processing Units (GPUs) serve highly efficient parallel execution for intensive computations at high-performance computing, deep learning, and embedded systems domains [1]. Developing efficient parallel programs for many specialized compute cores and hierarchical memory structures challenges programmers with traditional sequential programming backgrounds [2], [3]. The complex GPU hardware architecture and low-level parallel software models create a barrier to specializing in heterogeneous parallel programming skills. Teaching parallel programming for heterogeneous computing systems also becomes challenging for university educators due to limited access to parallel hardware resources and a lack of teaching materials for contemporary GPU devices.

There are efforts to adopt heterogeneous computing for undergraduate teaching materials [4], [5], [6]. While developing specific modules and adopting them in existing courses makes an introduction and increases the awareness of the students, the lack of details may be inadequate to expertise for the target topics. Additionally, the industrial support for university educators helps curriculum development in computer science departments [7]. In that case, the available content from different tracks needs to be studied and adapted according to the level and requirements of the students by blending main course materials.

In this paper, we demonstrate a teaching methodology for an accelerated computing course offered in the computer engineering department of a university adopted from the NVIDIA Deep Learning Institute (DLI) materials and present the results of the delivered contents by evaluating the impacts and possible improvements in future course offerings. The main contributions of our work are as follows:

- We design a computer engineering undergraduate elective course by blending NVIDIA DLI materials and an open-source deep learning inference library code.
- We utilize a wide range of hardware resources, from large-scale multiple GPU servers to small-scale embedded GPU devices. By utilizing GPU-based hardware platforms, the students can practice both basic CUDA programming and complex real-world applications, enabling them to work in different levels of software.
- For the evaluation of the course, we conduct a survey with questions to understand the students' impression of the course content and their feelings about further studies on the subject. Besides the survey results, we track the after-class activities by providing credits to additional self-paced courses.
- Our experience demonstrates that the students appreciate the blend of materials and feel satisfied with completing the course. Additionally, we observe that some of them prefer learning more about related topics by utilizing advanced teaching materials.

The remainder of this paper is organized as follows: Section II presents the main components of accelerated computing education, while Section III explains the NVIDIA DLI learning materials for accelerated computing. We present the adoption of existing teaching material using a mix of hardware and software platforms in Section IV and our evaluation results in Section V. Finally, we conclude the paper by summarizing our key findings in Section VI.

II. Accelerated Computing Education

Heterogeneous computing systems present high performance and less energy consumption by combining various device structures and configurations. Building heterogeneous systems by bringing together general-purpose multi-core processors (CPUs) and data-parallel graphics processing units

(GPUs) enables accelerated computing for efficient high performance in large-scale computing platforms [1].

Accelerated computing enables significant performance improvements by leveraging parallel hardware resources [8]. Understanding massively parallel execution and resource utilization in heterogeneous platforms with many-core GPUs requires expertise in low-level programming concepts and optimization strategies. While high-level directive-based programming models like OpenACC or OpenMP [9] offer accelerated computing in heterogeneous environments, fine-grained programming based on low-level programming models like CUDA [3] or HIP [10] enable more control in target executions to achieve performance improvements. Besides programming models, heterogeneous hardware platforms also have diverse characteristics by targeting dedicated computational operations. For instance, a deep learning inference model to run on an embedded GPU device has small memory and code size requirements, or a complex natural language processing engine requires programming multiple GPU devices by leveraging massive parallelism resources in single or multi-node large-scale environments.

For teaching heterogeneous computing, there are approaches to introduce parallel programming in different stages of undergraduate and graduate university courses [5], [6]. While dedicated elective courses offer details on parallel computer hardware platforms and parallel software development techniques [11], compulsory courses can also be designed to include parallel execution scenarios for the target environments.

III. Deep Learning Institute (DLI) Resources

NVIDIA DLI presents a wide range of learning resources to academia, from teaching kits to self-paced and instructor-led workshops, where the students may advance their knowledge in deep learning, accelerated computing, accelerated data science, graphics and simulation. For our *Heterogeneous Parallel Programming* course in the computer engineering department at Izmir Institute of Technology in Turkey, we utilize instructor-led accelerated computing workshops and accelerated computing teaching kit modules, and also enable students to enroll in self-paced courses. In this section, we summarize the content of the benefited materials while we explain our specific usage scenarios in Section IV.

A. Accelerated Computing Workshops

NVIDIA DLI offers instructor-led workshops in deep learning, accelerated data science, or accelerated computing [12]. DLI-certified instructors with experience in advanced topics of the workshop content deliver the workshops. The participants access fully configured, GPU-accelerated servers in the cloud to complete hands-on exercises included in the training. Consequently, they earn an NVIDIA Deep Learning Institute certificate in the course to demonstrate their theoretical and practical competency.

Instructor-led workshops contain five main topics: *Accelerated Computing*, *Data Science*, *Deep Learning*, *Generative AI and Large Language Models (LLMs)*, *Graphics and Simulation*. In our course, we utilize two *Accelerated Computing* workshops: *Fundamentals of Accelerated Computing with CUDA C/C++* and *Accelerating CUDA C++ Applications with Multiple GPUs*. Besides them, DLI offers *Fundamentals of Accelerated Computing with CUDA Python*, *Fundamentals of Accelerated Computing with OpenACC*, and *Scaling CUDA C++ Applications to Multiple Nodes* workshops on the same topic.

While instructor-led workshops are organized as paid online events, the *DLI University Ambassador Program* [13] enables qualified educators to teach free instructor-led courses for university students and researchers. By completing the multi-stage instructor certification process, educators affiliated with an academic institution are certified as *University Ambassadors* and become certified to teach the specific workshop. This program has several advantages: free DLI instructor certification, online teaching workshop materials, free software/hardware GPU resources, and financial support for travel and catering expenses for instructor-led workshops. The content of the adopted two workshops is explained as follows:

1) Fundamentals of Accelerated Computing with CUDA C/C++: The workshop teaches the basic CUDA programming techniques for accelerating C/C++ applications to run on parallel GPUs. The students learn to write basic CUDA code, configure code parallelization with CUDA threads, and memory copy operations between the CPU and GPU device. Additionally, they profile CUDA code with NVIDIA Nsight Systems performance profiling tool [14] and gain experience in code optimization techniques based on visual profiling analysis. As the final task of the workshop, they parallelize and optimize the serial particle simulation code on target GPU system.

2) Accelerating CUDA C++ Applications with Multiple GPUs: The workshop covers how to write efficient CUDA C++ programs that efficiently and correctly utilize multiple GPUs in a single node, significantly improving the performance of the programs on systems with multiple GPU devices. It introduces concurrent CUDA streams to overlap memory transfers with GPU computation on a single GPU, then extends to utilize multiple GPUs on a single node to scale workloads across all available GPUs.

B. Accelerated Computing Teaching Kit Modules

To help university educators incorporate accelerated computing into their courses, DLI offers downloadable teaching kits that include course materials that were co-developed with different university faculties [15]. Each kit, freely available for instructors, includes editable lecture presentations and hands-on lab exercises with sample solutions. The instructors can use the teaching kits in the courses by adapting available material. *Accelerated Computing Teaching Kit* includes several modules such as *Introduction to CUDA C*, *CUDA Parallelism Model*,

Memory and Data Locality, *Parallel Computation Patterns*, and *Memory Access Performance.*

C. Self-Paced Courses

Besides instructor-led workshops, DLI offers online self-paced courses, where registered students follow the online materials from NVIDIA infrastructure on their own and receive certificates upon successful completion. Like instructor-led workshops, there are different-level courses on different topics, including *Accelerated Computing*.

IV. Course Structure

In the computer engineering department at Izmir Institute of Technology in Turkey, we offer *Heterogeneous Parallel Programming* course based on NVIDIA hardware and software resources and teaching materials. The semester-long technical elective course for senior-level undergraduates provides GPU-based heterogeneous programming fundamentals. It covers the basic concepts of parallel architectures and parallel programming, CUDA programming model topics, and practical examples by presenting the fundamentals of accelerated computing with CUDA. The course objectives are as follows:

- Understand CPU-GPU heterogeneous architectures,
- Design and implement heterogeneous parallel programs,
- Understand GPU execution units and memory hierarchy to foster execution performance,
- Understand performance evaluation and optimization methods targeting GPU devices.

In our course, we utilize *NVIDIA-DLI* workshop topics [12], *Accelerated Computing* teaching kit modules, and *NVIDIA-BTF* resources and offer a complete learning flow for heterogeneous programming on NVIDIA devices. To introduce basic parallelism and CUDA topics and also for advanced performance lectures, we adopt the relevant teaching kit modules. As a university ambassador and a DLI-certified instructor, we organize two workshops (as given *DLI-1* and *DLI-2* in Table I) as part of our in-class lectures and deliver the content for six weeks of the semester. The students register for the workshops beforehand and have access to the course material and remote GPU resources during the workshops. We review workshop content during lecture hours by explaining more advanced topics separately and let the students work on hands-on programming practices by guiding them. After spending time on programming problems, we run the solution and discuss the execution with possible performance considerations. With both command line and visual profiling tools, the students experience CUDA code development and analyze the performance impacts of the code modifications on the target execution. Finally, for a hands-on experience on embedded GPU devices, we utilize Jetson Nano devices, received with a teaching project grant, and the relevant open-source code examples.

Table I shows our course schedule with course topics and corresponding components provided by NVIDIA teaching resources and hardware grant. Specifically, we start the semester by introducing the course content and basic parallelism concepts (based on Teaching Kit *Module 17 - Computational Thinking For Parallel Programming*). After explaining the fundamental CUDA topics based on Teaching Kit *Module 2 - Introduction to CUDA C*, we deliver DLI Workshop *Fundamentals of Accelerated Computing with CUDA C/C++* (given as *DLI-1* in Table I) for three weeks, where the students learn theoretical CUDA threading concepts and complete the workshop programming tasks at remote GPU hardware platforms. Additionally, we cover DLI Workshop *Accelerating CUDA C++ Applications with Multiple GPUs* (given as *DLI-2* in Table I) for another three weeks after covering low-level CUDA performance topics provided in Teaching Kit *Module 6 - Memory Access Performance*. Finally, we introduce Jetson Nano Developer Kit, provided by the *NVIDIA-BTF* grant, and the students work on and complete an object detection project based on Jetson AI Lab source code [16].

TABLE I
Weekly Course Topics.

Course Topic	NVIDIA Component
Introduction	-
Parallelism	Teaching Kit Module 17
Introduction to CUDA	Teaching Kit Module 2
Accelerated Computing with CUDA C/C++	DLI-1
CUDA Performance	Teaching Kit Module 6
Multiple GPUs	DLI-2
Jetson Inference	Jetson AI Lab

In the following sections, we explain the details of each component and how we adapt the available software and hardware resources and teaching material to our course content.

A. Introduction Lectures

We start the semester by reviewing the basic parallel computer architecture topics based on the summary of the concepts in the Hennessy&Patterson book [8]. Then, we cover *Module 17 - Computational Thinking For Parallel Programming* content by discussing parallel programming fundamentals and heterogeneous parallel computing benefits. We introduce GPU architecture and accelerated computing on GPU devices based on the slides from *Module 2 - Introduction to CUDA C*. The introductory lectures mainly provide the theoretical CUDA basics with sample codes but do not include practical hands-on programming practices.

B. Accelerated Computing with CUDA C/C++

As the formal introduction to CUDA programming model with practical example code, we offer the *Fundamentals of Accelerated Computing with CUDA C/C++* workshop. We deliver the workshop for our students as a three-week course material. With a background in C programming language and computer architecture topics, the students adapt the remote workshop infrastructure based on Jupyter notebooks and develop and run the programming practices. While the students run the given code at remote resources and see the output of the executions, they also generate profile results based on NVIDIA Nsight Systems profiler. Not only do they analyze

the command line output, but they also track GPU activities and CPU-GPU interactions from the Systems visualization interface, where they can access remote desktops. As one of the programming assignments defined as part of the course, they are required to complete the final task of the workshop and earn the certificate.

C. CUDA Performance Lectures

After the first workshop, *Fundamentals of Accelerated Computing with CUDA C/C++*, we cover advanced CUDA performance optimization techniques. We define parallel performance metrics *flop rate* and *throughput* received from GPU executions by illustrating compute-bound and memory-bound application examples. Then, we present performance bottlenecks by discussing global memory bandwidth and warp scheduling issues based on the content at *Module 6 - Memory Access Performance*. At the end of the performance lecture weeks, the students are trained in both basic and advanced CUDA programming techniques on a single GPU system.

D. Multiple GPUs

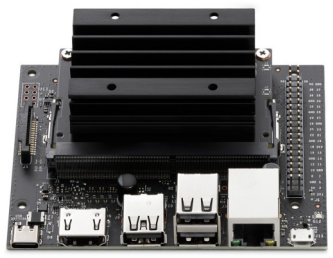
As the second DLI workshop, we offer *Accelerating CUDA C++ Applications with Multiple GPUs* workshop. We deliver the workshop as a three-week course material similar to the previous one. The students experience CUDA programming on multiple GPU devices using a remote hardware platform with four V100 NVIDIA GPUs. Passing the final task of the workshop, which includes multiple choice questions, is defined as an assignment in the course.

E. Deep Learning Inference with Jetson Nano Developer Kit

After studying fundamental CUDA concepts with server-scale GPU devices at remote resources, the students work with embedded GPU devices for a practical AI workload. Thanks to Jetson Nano Developer Kits, we extend our course by including a hands-on deep learning project experience. Firstly, we introduce Jetson Nano developer kits, then give the basics of the deep learning inference process. Finally, we let the students work on an object detection code that is available as an open-source project. We summarize the final project steps as follows:

1) Introduction to Jetson Nano: NVIDIA Jetson devices are small and powerful computers for embedded applications [17]. Figure 1 presents the NVIDIA Jetson Nano 2GB developer kit and its technical specifications. While there are more modern versions (such as the Jetson Orin Nano and Jetson Xavier series) of the Jetson devices, we use the NVIDIA Jetson Nano 2GB developer kit for our teaching purposes.

Before letting the students work on the final project, we introduce Jetson hardware and software features. After explaining technical specifications, we give a brief introduction to the system software. NVIDIA JetPack SDK [18] provides a full development environment for powering the Jetson modules and building accelerated programs. Jetson Stats [19] is a package for monitoring and controlling the



TECHNICAL SPECIFICATIONS

GPU	128-core NVIDIA Maxwell
CPU	Quad-core ARM A57 @ 1.43 GHz
Memory	2 GB 64-bit LPDDR4 25.6 GB/s
Storage	microSD (Card not included)
Video Encode	4Kp30 | 4x 1080p30 | 9x 720p30 (H.264/H.265)
Video Decode	4Kp60 | 2x 4Kp30 | 8x 1080p30 | 18x 720p30 (H.264/H.265)
Connectivity	Gigabit Ethernet 802.11ac wireless*
Camera	1x MIPI CSI-2 connector
Display	HDMI
USB	1x USB 3.0 Type A, 2x USB 2.0 Type A, 1x USB 2.0 Micro-B
Others	40-pin header (GPIO, I²C, I²S, SPI, UART) 12-pin header (Power and related signals, UART) 4-pin Fan header*
Mechanical	100 mm x 80 mm x 29 mm

* Not initially available in all regions

Fig. 1. NVIDIA Jetson Nano 2GB developer kit and its technical specifications.

NVIDIA Jetson devices. While the Jetson Nano devices are pre-installed with NVIDIA JetPack SDK software, the students are required to install the Jetson Stats package. They work with JetPack 4.6.1, which includes CUDA 10.2, cuDNN 8.2.1, TensorRT 8.2.1, and NVIDIA Nsight Systems 2021.5 software versions.

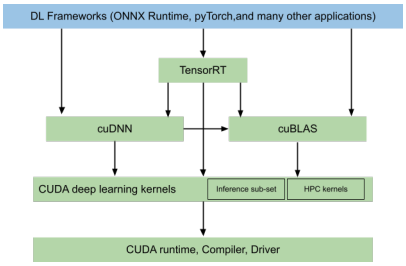


Fig. 2. NVIDIA inference stack.

2) Introduction to Deep Learning Inference: We give brief information about artificial intelligence and deep learning with an emphasis on the requirement of intensive computing resources and time. Specifically, we focus on the *NVIDIA Inference Stack* inference process (given in Figure 2) and *TensorRT* framework [20], which includes a deep learning inference optimizer and runtime that delivers low latency and high throughput for inference applications.

3) Hands-On Object Detection Example: As an illustrative example, we utilize an open-source object detection code [16] based on *TensorRT* and a pre-trained *DetectNet* network. After downloading the available code from the GitHub repository, the students are first required to run the Docker image to understand the execution flow. Then, they build the project from the source code, execute it on Jetson Nano, and profile the executable with the *nvprof* profiler. In order to understand the accuracy and performance differences, they work with

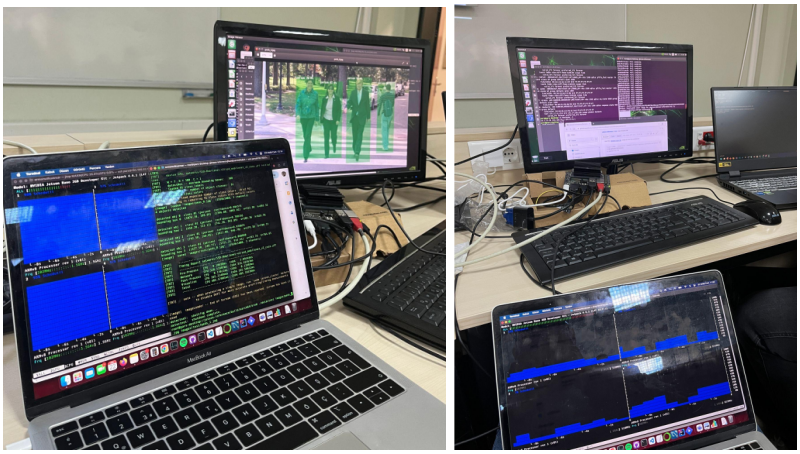


Fig. 3. Working object detection code with Jetson Nano developer kit (photo courtesy of Erman Utku Avşar).

different networks, including *ssd-mobilenet-v2*, *ssd-inception-v2*, and *peoplenet* for all available images in the repository. They collect *SM Efficiency* metric for CUDA kernel functions in the execution of the target networks and identify the most and the least efficient kernel functions. Figure 3 presents photo samples taken in the classroom while the students are executing the object detection code on Jetson Nano devices, visualizing the output on the screen, and profiling the target execution. While the students do not delve into details by examining the differences among kernel function codes, they gain a notion of performance evaluation for a small-scale real-life application.

F. Self-Paced Courses

As part of the course, we offer financial support for the students to enroll in NVIDIA-DLI self-paced courses and earn certificates. We do not define any formal assignment for that part, so the students can follow any course on any track. The preferred courses are from diverse topics. Those courses enable the students to learn more about the specific topics they are interested in, including accelerated computing, data science, or deep learning concepts. For this additional and optional part, there were 90 course enrollments from a wide-range of topics including but not limited: *Getting Started with Deep Learning, Accelerating End-to-End Data Science Workflows, Fundamentals of Accelerated Computing with OpenACC, RAPIDS Accelerator for Apache Spark, High-Performance Computing with Containers, Generative AI with Diffusion Models*. Among those enrollments, the students received 25 certificates.

V. Course Evaluation

Our semester-long *Heterogeneous Parallel Programming* elective course is taken by 40 undergraduate students from the computer engineering department and 3 graduate students from the computational science and engineering program at Izmir Institute of Technology. We conduct the weekly lectures in fully-equipped PC labs with high-speed internet connection. While 7 students failed the course, 10 undergraduate students got the highest letter grade (i.e., *AA*) by completing all assignments and receiving the workshop certificates.

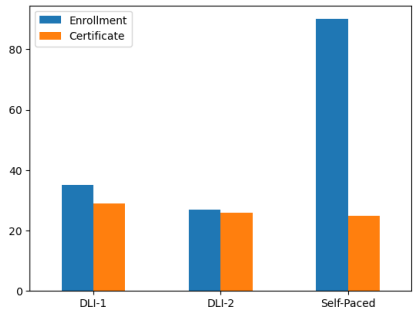


Fig. 4. The number of DLI course enrollments and received certificates.

Figure 4 presents the number of students who enrolled for instructor-led DLI workshops and self-paced courses and received certificates for the course. For instance, while 35 students were enrolled in the first workshop, *Fundamentals of Accelerated Computing with CUDA C/C++*, 29 of them received certificates by completing the final task. Among 33 students who passed the course, we can see that some of them could not get the compulsory certificates from instructor-led workshops (i.e., *DLI-1* and *DLI-2*). However, for those who could not register for the instructor-led workshops and receive the certificate, there was an option to register for the lightweight self-paced courses and receive the certificates for the corresponding courses. It is clear that there were students taking this option due to missing the instructor-led workshops.

The DLI workshops encourage the participants to fill in a feedback form after they complete the courses. Figure 5 presents the results of the feedback form on the learning experience, course meeting expectations, and instructor satisfaction levels on different criteria. The *DLI-1* scores on the learning experience questions like *Registration* and *Navigation* are lower than *DLI-2* due to lack of platform experience. We can see that the students felt more comfortable in the second workshop after they got used to the workshop registration

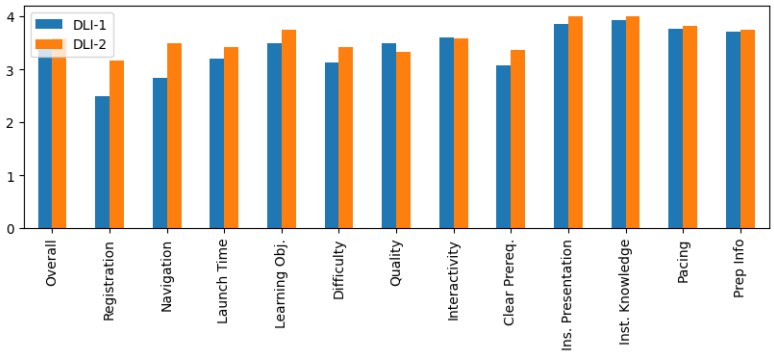


Fig. 5. Results of the feedback form on the learning experience from DLI courses.

TABLE II
OUR COURSE EVALUATION SURVEY RESULTS (1:NOT GOOD/CERTAINLY NO, 4:VERY GOOD/CERTAINLY YES.

Question	Rate 1	Rate 2	Rate 3	Rate 4
Support received from the instructor	0	0	3	16
Confidence to do more advanced work in the subject	0	0	6	13
Increase your interest in the field of study	0	2	5	12
The overall experience of the course	0	1	5	13
Recommend this course to a friend	0	0	3	16

process and navigation interface in the first one. Other than that, the overall feedback on the workshop satisfaction is quite high for both workshop delivery.

The Izmir Institute of Technology (IZTECH) requests the students to fill out a course evaluation survey for each course at the end of each semester. For our course, 39 students out of 43 participated in the survey and answered some defined questions. Figure 6 presents the results for three relevant questions about the course content. Both the examples and the assignments are appreciated by approximately 3/4 of the students.

Since the IZTECH course evaluation survey is generic for all departments and courses, specifically for our course, we designed and conducted an additional survey by considering the specific course topics and student feedback. 19 students participated in the non-compulsory survey and answered the 7 questions. Two of our questions (given in Figure 7) were about the course activities, and five of them (given in Table II) were more general course evaluation and feedback.

The results given in Figure 7 demonstrate that the *Fundamentals of Accelerated Computing with CUDA C/C++* workshop (*DLI-1*) has been evaluated as the most helpful to understand the concepts. Since the workshop introduces the main CUDA programming techniques with its practical programming task, the students were easily engaged during those weeks. Both *Introduction* and *CUDA Performance* lectures have been evaluated as the least helpful due to lack of details. On the other hand, *Jetson Nano* lectures have been evaluated as the most helpful for 20% of the students, while 30% of the students specified them as the least helpful. Since the Jetson Nano developer kit installation and inference execution steps require Linux environment experience, the students, who haven't taken an Operating Systems course or have no hands-on Linux experience, had difficulty following the guidelines and completing the steps. Although we have not investigated the reason further, we think that the Jetson Nano lectures might have been more helpful if the students had prior Linux knowledge. Despite this, 20% of the students motivated us to spend more time in related activities.

As part of the survey, we requested the students to rate the remaining question topics (rates from the most negative (1) to most positive (4)), and Table II presents the questions and the number of students evaluated it as the given rate. While more than 80% of the students think that they received support from the instructor and recommend the course to others, the overall experience exhibits a lower rate (around 70%). On the other hand, most of the students (around 65%) have confidence and interest in the course topics, the rates are relatively low compared to the other questions. While the students could follow the course content and complete the given tasks, they do not feel they can achieve more in accelerated computing. Introducing both academic and industrial real-life examples and guiding them to perform more advanced work might have been helpful to feel more confident in the concepts and increase their interest in practical application areas. Inviting guest lecturers with academic or industry experience might have given more specific examples.

VI. SUMMARY

Including single and multiple GPU programming concepts from DLI workshop contents, delving into parallel performance issues based on the Teaching Kit materials, and a hands-on inference project on an embedded system, Jetson Nano, the course has been a great success for undergraduate students. The combination of diverse accelerated computing teaching

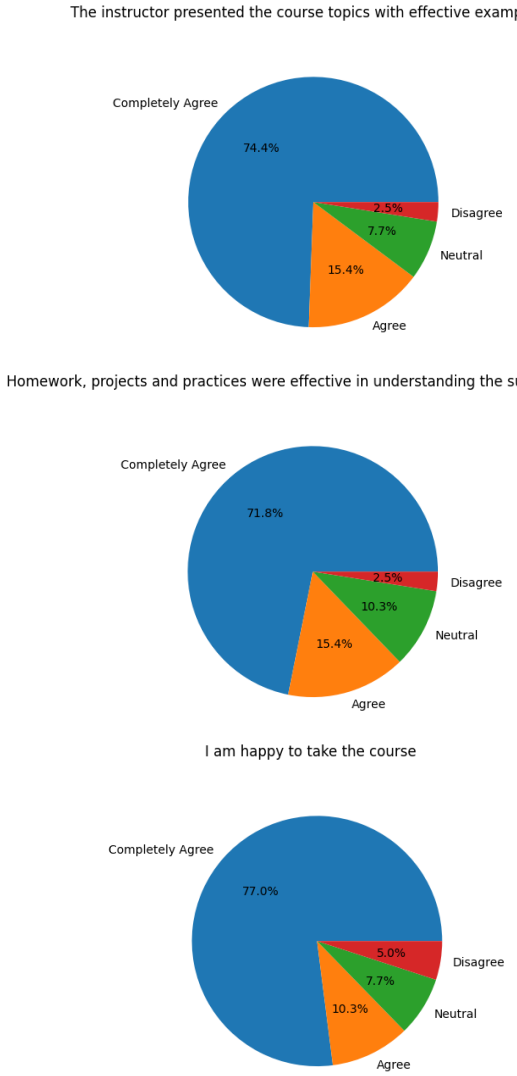


Fig. 6. IZTECH course evaluation survey results.

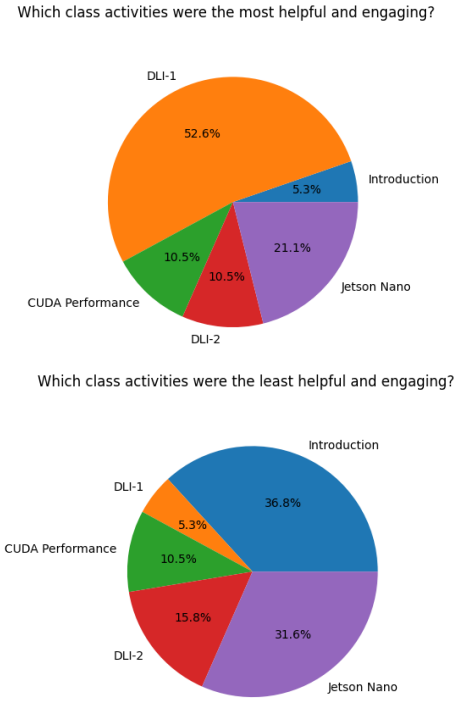


Fig. 7. Our course evaluation survey results.

materials and resources for an undergraduate course has been successfully implemented and appreciated by the students.

Motivated by the students' feedback to extend the work to multi-node GPU systems, the instructor registered for DLI workshop *Scaling CUDA C++ Applications to Multiple Nodes* and has been certified to teach the workshop after a successful evaluation in March 2024. We plan to include that workshop content in our course offering next year.

ACKNOWLEDGMENT

This curriculum was supported by a Turkey Earthquake Relief Program grant from NVIDIA and BTF and utilized NVIDIA Jetson Nano 2GB developer kits. The authors are thankful for the support of NVIDIA Deep Learning Institute.

REFERENCES

[1] T. M. Aamodt, W. W. L. Fung, T. G. Rogers, and M. Martonosi, *General-Purpose Graphics Processor Architecture*. Morgan and Claypool Publishers, 2018.

[2] P. Hijma, S. Heldens, A. Sclocco, B. van Werkhoven, and H. E. Bal, "Optimization techniques for gpu programming," *ACM Comput. Surv.*, vol. 55, no. 11, mar 2023.

[3] W. H. Wen-Mei, D. B. Kirk, and I. El Hajj, *Programming Massively Parallel Processors: A Hands-on Approach*. Morgan Kaufmann, 2022.

[4] D. P. Bunde, K. Ahmed, S. Ayloo, T. Brown-Gaines, J. Fuentes, V. Jatala, R. Kurniawati, I. Öz, A. Qasem, P. J. Schielke, M. C. Tedeschi, and T. Y. Yeh, "Adopting heterogeneous computing modules: Experiences from a touch summer workshop," in *IEEE/ACM International Workshop on Education for High Performance Computing, EduHPC 2022, Dallas, TX, USA, November 13-18, 2022*. IEEE, 2022, pp. 18–25. [Online]. Available: https://doi.org/10.1109/EduHPC56719.2022.00008

[5] A. Qasem and D. P. Bunde, "Heterogeneous computing for undergraduates: Introducing the touch module repository," in *SIGCSE 2022: The 53rd ACM Technical Symposium on Computer Science Education, Providence, RI, USA, March 3-5, 2022, Volume 2*. ACM, 2022, p. 1201.

[6] A. Qasem, D. P. Bunde, and P. Schielke, "A module-based introduction to heterogeneous computing in core courses," *J. Parallel Distributed Computing*, vol. 158, pp. 56–66, 2021.

[7] "Nvidia deep learning programs for educators." [Online]. Available: https://learn.nvidia.com/en-us/training/educator-programs

[8] J. L. Hennessy and D. A. Patterson, *Computer architecture: a quantitative approach*. Elsevier, 2011.

[9] R. Chandra, L. Dagum, D. Kohr, R. Menon, D. Maydan, and J. McDonald, *Parallel programming in OpenMP*. Morgan kaufmann, 2001.

[10] "Amd rocm documentation." [Online]. Available: https://rocm.docs.amd.com/

[11] B. Gyires-Tóth, I. Öz, and J. Bungo, "Teaching accelerated computing and deep learning at a large-scale with the nvidia deep learning institute," *Journal of Computational Science*, vol. 14, no. 1, 2023.

[12] "Nvidia deep learning programs for educators." [Online]. Available: https://learn.nvidia.com/en-us/training/instructor-led-workshops

[13] "Nvidia university ambassador program." [Online]. Available: https://www.nvidia.com/en-us/training/educator-programs/university-ambassador-program/

[14] "Nvidia nsight systems." [Online]. Available: https://developer.nvidia.com/nsight-systems

[15] “Nvidia dli teaching kits for educators.” [Online]. Available: https://www.nvidia.com/en-us/training/teaching-kits/
[16] “Hello ai world guide to deploying deep-learning inference networks and deep vision primitives with tensorrt and nvidia jetson.” [Online]. Available: https://github.com/dusty-nv/jetson-inference
[17] “Nvidia jetson nano.” [Online]. Available: https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/jetson-nano/product-development/
[18] “Nvidia jetpack sdk.” [Online]. Available: https://developer.nvidia.com/embedded/jetpack/
[19] “Nvidia jetson stats.” [Online]. Available: https://developer.nvidia.com/embedded/community/jetson-projects/jetson_stats/
[20] “Nvidia tensorrt sdk for high-performance deep learning inference on nvidia gpus.” [Online]. Available: https://github.com/NVIDIA/TensorRT