Teaching practice reforms towards software-hardware collaboration in computer system ability training-Taking FPGA Design course as an example

Ying Li[1] Jianwei Niu[1] Simbarashe Matutu[1] Qianben Qi[1]
liying@buaa.edu.cn
[1]School of Computer Science and Engineering, Beihang University, Beijing, China

***Abstract*—Computer system ability training is a new trend in computer education. This paper proposed an innovative experimental teaching method of software and hardware collaborative design to better develop students' system view, structure view, engineering view of computers. An FPGA-based CNN accelerator was designed to combining the knowledge from software to compilation and then to hardware. The main innovations are: ① Innovation of experimental system: A "curriculum tree" based on knowledge map was built to identify the implicit relationship between software and hardware knowledge. The curriculum was changed from Horizontal Teaching to Vertical Teaching in order to reduce the difficulties of cultivating system ability; ②Innovation of experiment contents: it proposed an experiment teaching strategy of "Managing Complexity With Simplicity" guided by Occam's razor and used some effective methods to simply the system knowledge of each course around the top-level goals; ③Innovation of experiment methods: a procedural and flow-based experiment model based on hierarchical experimental contents was used to achieve spiral progressive learning from software design to hardware simulation and then to system development; ④Innovation of experiment platforms: an innovative method of conducting experiments, MOOE (MOOE = Experiment + MOOC) was proposed to allow students to do experiments "anytime, anywhere and on demand".**

Keywords—Software-Hardware Collaboration, Vertical Teaching, MOOE, FPGA-based Accelerator

I. OVERVIEW

In the latest publication of CS2013 released by ACM/IEEE, 4 new knowledge domains involving system-level content design were added namely System Fundamentals (SF), Parallel and Distributed computing PD, Platform-based Development (PBD), Information Assurance and Security (IAS). [1] It can be observed that the focus of ACM/IEEE CS2013 is to emphasize the cultivation of system knowledge and system capabilities. It believed that computer science graduates should have a system-level view and need to think abstractly from multiple levels of details. Their understanding of computers should go beyond the implementation details of various components, rather it should think from the overall view of computer system structure and construction analysis process.

In response, in January 2017, China's Ministry of Education launched the research framework of "Emerging Engineering Education" (3E), which aimed at coping with the challenges of a new round of scientific and technological revolution and industrial changes. Building a new curriculum pattern of "Integration of multidisciplinary, Integration of general education and professional education, Integration of education and industry" meant to cultivate students' interdisciplinary talents with stronger engineering practice skills and interdisciplinary characteristics. The "Emerging Engineering Education" embodies the idea of the "Big Engineering View", which encourages higher education to return to the standard of "learning for application and knowledge for action". The "Big Engineering View" was first put forward in the report "Engineering with a *Big*E: Integrative Education in Engineering" issued by MIT in 1994. [2] It emphasized the need to provide a "large" scale of basic scientific knowledge and a "large" scale of engineering practice ability for large scale complex projects and it aimed to break the scope of traditional engineering science knowledge through interdisciplinary integration.

Given the new developments in of computer higher education, the Teaching Steering Committee of the Ministry of Education for Higher Education in China, through sorting out the 12 abilities required for graduation from engineering education certification, believed that senior computer professionals should have the following abilities: computational thinking skills, algorithm analysis and design skills, programming design and implementation skills. System capabilities subcategory, which included various system skills are particularly important, accounting for 75% of the total capabilities. (see Table 1) [3]

Table1 Distribution of computer capabilities

Skill composition		Skill points
Computational thinking ability		9
Algorithm design and analysis ability		8
Programming and implementation capabilty		3
System Capabilty	System cognition	6
	System design	19
	System development	23
	System application	14

II. SOFTWARE-HARDWARE COLLABORATION TEACHING

The implications of system capability attest to the fact that cultivation of system capability can only be achieved through a large number of practical courses.

A. Innovation of experimental system

To help students establish a complete view of the computer system, we use "to design a software-hardware integrated environment that integrates deep learning models, deep learning compiler tool-chains, and FPGA hardware accelerators", as the top-level practical teaching goal. Around this center, we establish an integrated experimental system of software and hardware, which involves multiple courses such as program design, compiler principles, and computer organization. Therefore, it is necessary to conduct a fine-grained analysis and research on the relationship, level of difficulty, and order of teaching among the knowledge points of each course.

For this reason, a knowledge map is used to visually display the software and hardware knowledge points involved in the cultivation of computer system skills. It's in the form of a knowledge network and establishes a hierarchical and classified knowledge system. Specifically, according to the construction of knowledge set in curriculum group, the paper studies the extraction methods of entity, attribute and entity-relationship in teaching text data such as structured (questionnaire, table data), semi-structured (syllabus, teaching calendar) and unstructured (teaching materials, courseware, homework), and puts forward a method of constructing a "curriculum tree" which will be implemented based on the knowledge map. The "curriculum tree" describes the hierarchical relationship between the courses in the curriculum group, and each node in the tree represents a core course of the curriculum group, as shown in Figure1. This method of knowledge map construction is based on multi-source heterogeneous teaching data breaks the subjective analysis methods which mainly rely on the experience of teachers to design teaching content. It makes full use of artificial intelligence and other information methods to deeply explore the hidden relationship and implicit characteristics between knowledge points. It can not only avoid the generation of knowledge islands but also effectively support the implementation of "dynamically generative" teaching strategies. Knowledge points can be adjusted in time according to the learning situation of students, and "open, dynamic and flexible" teaching content can be created.

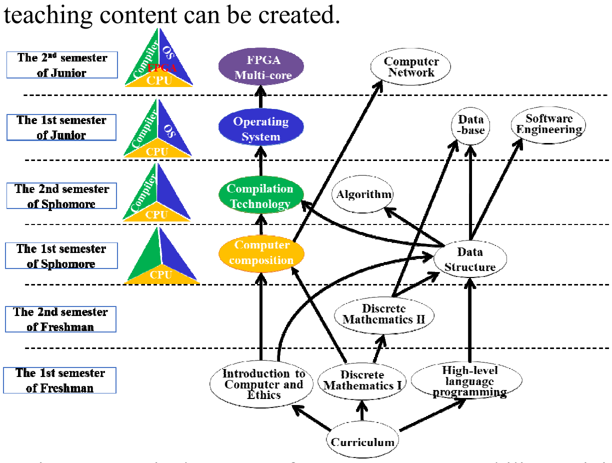


Figure 1 Curriculum Tree of Computer System Ability Training in Beihang University

1) To optimize the experimental curriculum system

The traditional teaching model of software and hardware teaching cuts the knowledge structure into several independent curriculum units and the boundaries between courses are clear but, it weakens the relevance and systemicity of the knowledge system. The optimization of the curriculum for software hardware collaboration is a process of repeated iteration and continuous tuning, which requires fine-grained analysis and research on the relationship, difficulty degree and teaching order among the knowledge points of each course in the curriculum. Due to the diversity of teaching data sources, isomerization of forms and differentiation of contents, the development of the curriculum faces great challenges such as the need for a large investment, long cycle and difficult implementation. Therefore, it is necessary to use knowledge mapping to identify the implicit relationship and internal logic among disciplines, courses and software/hardware knowledge. Through the integration and optimization of knowledge points, the curriculum can be reversely designed according to the teaching objectives, and a knowledge system matrix with reasonable structure and coherent content is to be formed. Eventually, the curriculum will be transformed from being a "horizontal division based on the curriculum" into being a "vertical division based on the relevance and causality of the cognitive system ", as shown in Figure 2, which can effectively reduce the difficulty of cultivating a computer system view.

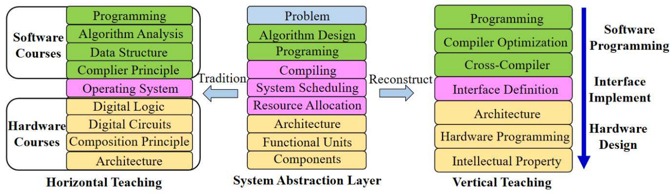


Figure 2 From Horizontal Teaching to Vertical Teaching

(Note: Green, pink and orange parts refer to software, interface and hardware, respectively)

2) To optimize the experimental curriculum design

According to educational theories, the ordering of knowledge is a prerequisite for delivering curriculum content. The content of the course generally includes two types, these are declarative knowledge of "why and what is" (explicit knowledge such as concepts and principles) and procedural knowledge of "what and how" (recessive knowledge such as methods and techniques). Obviously, the ordering and integration of these two types of knowledge must not be a simple "algebraic sum", but an integrated "vector sum". It needs to be done by educators under the guidance of decision-making rules. However, the researchers, organizers, and implementers involved in the development of these rules and standards are all influenced greatly by the traditional discipline system. This leads them to have obvious subjective bias and inherent consciousness when planning the curriculum system. Therefore, it is necessary to establish an objective decision-making mechanism based on the knowledge map, which can accurately extract temporal relationship and structural characteristics between knowledge points, and automatically learn the fusion rules of knowledge in a data-driven manner. Under the premise that the total amount of knowledge remains unchanged, then form intelligent ordering of knowledge can both improve the system completeness and logic of teaching design and finally achieve the effect of rudimentary learning such as "1+1>2".

B. Innovation of experiment contents

Hardware courses are difficult for students to learn. In order to solve this problem, the course group proposed an experiment teaching strategy of "Managing Complexity With Simplicity" guided by Occam's razor and used some effective methods such as separating individual problems, decoupling software and hardware design to simply and optimize the knowledge system of each course around the top-level goals.

When designing, we divide the teaching content into three parts with clear boundaries and close related-context based on the hierarchy and modularization design idea of IOS seven-layer network structure. Specifically, from the top-down approach, the three parts are software development, cross compiling, hardware designing. And it ensures that each layer can perform its duties and it is responsible for a specific function and provides standardized interfaces between each layer. This makes it easy to expand and maintain the teaching content.

The experiment uses the MNIST handwritten digit image set, and designs an acceleration method to achieve handwritten digit recognition through a software-hardware coordination mechanism on a FPGA platform, so as to obtain the best recognition accuracy and speed. Specifically, the experiment adopts the method of software-hardware collaborative computing to construct the upper, middle and lower three-tier architecture of "PC + ARM + FPGA": the upper layer uses Python language to independently design a lightweight VGG16 neural network model (VGG16 Simple) to train MNIST data set (TensorFlow and other architectures cannot be directly used). It uses the TVM deep learning compiler to convert the trained VGG16 network into a bitstream file that can be run on FPGA. The middle layer uses a specific communication protocol to burn the bitstream file generated by the upper layer to the lower layer. The lower layer uses the Verilog hardware description language to implement the trained VGG16 network on the FPGA (Xilinx ZYQN7100). The heterogeneous platform of "PC + ARM + FPGA" is built to give full play to their advantages to obtain the best acceleration effect in low-power scenarios. PC is responsible for VGG network training, ARM is responsible for network compilation and data communication, FPGA is responsible for high-performance computing. [8]

The vertical and hierarchical experimental contents designed in this paper enables students to both understand computer hardware design from the perspective of programmers, and understand the principle of program execution from the perspective of hardware designers thus, cultivating a deep understanding of the working mechanism of software and hardware collaboration. This greatly simplifies the system design process and enables them to build the entire computer system design process from software to compilation and then to hardware.

C. Innovation of experiment methods

To strengthen the training of computer system skills, this paper proposed a progressive teaching method of "hardware-software collaboration, from software to hardware, from hardware to software". The course focuses on designing a CNN for handwritten digit recognition and it covers the whole development process from software to hardware, from algorithm design to system design and from design to testing. The course uses a multi-dimensional comparative analysis method to compare the similarities and differences of the same CNN (VGG16) in different execution environments (CPU and FPGA). This is done from the perspectives of the design principles, implementation methods and execution efficiency, which helps the students understand the interaction between software programmability and hardware programmability, thus gradually form the three-dimensional thinking mode required for system development. The specific methods used are as follows:

1) To set up hierarchical experiment content

Hierarchical experimental content can be used to achieve spiral progressive learning from software design to hardware simulation and then to system development. The experimental projects are divided into different levels such as basic, comprehensive, design and research as shown in Table 2. The early stage of the course is made up of basic experimental contents, which mainly focus on allowing students to master the experiment environment and basic operating methods. The middle of the course is comprised comprehensive experimental contents and students need to comprehensively use their hardware knowledge and programming skills they have mastered in the previous stage to complete complex CNN accelerator on FPGA platform. They will need to make qualitative and quantitative analysis of the experimental process and results. The later stage of the course is made up of design experimental content, which is a relatively higher level of experimental contents. It is not only both a summary and consolidation of theoretical study, but also a training for practical ability and engineering skills. The follow-up to the course is a research experiment. After a semester of investigations, students who have certain academic research capabilities and are interested in the related topics are selected for long-term and in-depth training.

2) To adopt a procedural and flow-based experiment model

The experiments are carried out as a team project and divide students into groups. The experiment process is subdivided into "experiment literature preview, teamwork operation, and experiment report in the form of a discussion and experiment assessment in the form of oral defense. At the same time, each experiment stage is standardized to archive process monitoring from data preview, content design, group discussion, result verification, report writing, etc., as shown in Figure 3.

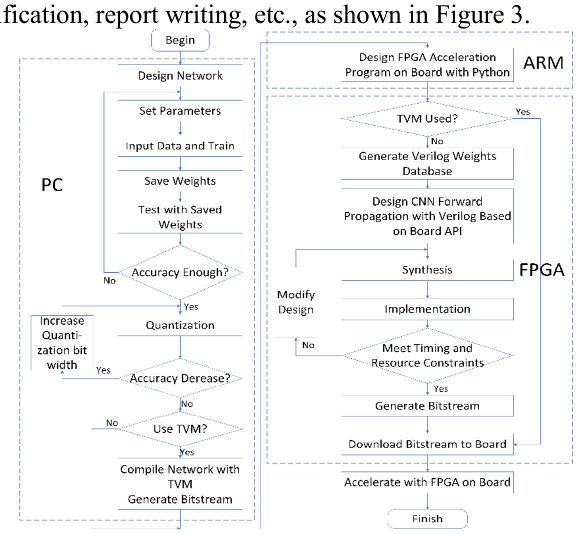


Figure 3 Work Flow Chart of CNN accelerator based on FPGA

D. Innovation of experiment platforms

In order to allow students to do experiments "anytime, anywhere and on demand", we propose an innovative method of conducting experiments, MOOE (Massive Open Online Experiments), MOOE = Experiment + MOOC. It is an organic combination of "Internet + education + platform" to create a laboratory environment that can be operated remotely. MOOE is highly advanced in terms of platform design and teaching concepts. It emphasizes "the authenticity, credibility and integrity of the experiment process using the internet to replicate a lab environment. [9][10] It is manly realized by: ① Designing a hybrid experiment model to achieve seamless integration of fully-physical operations and semi-physical simulations and modeling simulations. ②Construction of an online experiment teaching platform with characteristics of "full time and space, follow-up, and hardware cloud" and it supports online use of experimental resources, online operation of experimental equipment and online guidance of the experimental process, as shown in Figure 4. ③Breaking the tight coupling relationship between the traditional experiment platform and experimental equipment and fragmenting the experiment contents to realize fragmented segmentation learning; ④Using information technology, the whole process of teaching, learning, evaluation, testing, and practice is realized with the help of the internet, which makes for seamless connection and asynchronous interaction between theoretical teaching and experiments. MOOE enable the tracking of experiment progress and logging query functions. Teachers can know the behaviour of students doing online experiments at any time, including completion time, number of submissions, evaluation results etc. At the same time, MOOE also provides online Q&A functions through which students can directly communicate and interact with teachers. In short, MOOE promotes the idea of "experiment follows people" which completely solves the problem of time, space and resource constraints in teaching experiments, as shown in Figure 4.

1) To improving the integrity of the teaching content

The curriculum in MOOE is broad in nature. It replicates the entire lab teaching process over the internet. It not only implements online teaching in the curriculum, but also provides online use of experiment equipment, online operation of the experiment processes and online guidance to the experiment content. A series of highly interactive functions such as online display of experiment results, online learning of the complete teaching process and closed loop teaching from theoretical learning to practical training to ability training is implemented. Thus, making teaching and learning both well done with online mode only. It enables learners to interact with laboratory teachers, lab equipment and lab classmates online to complete various experiments collaboratively, as shown in Figure 4. MOOE effectively solves the current situation of separate teaching, learning, and training. It enables students to directly carry out supporting practical training after gaining knowledge online, thus realizing the transition from a learning mode based on time control to a learning mode based on personal ability.

2）To improve the sharing of teaching resources

The main advantage of MOOE lies in resource sharing. Traditional experiment teaching methods generally pays attention to the sharing of physical resources, that is, increasing the utilization rate of resources by extending hours thee lab will be open and building shared platforms. This kind of sharing has many limitations in terms of scale, type, objects, and effects. MOOE makes full use of advanced computer technology and internet technology, combines the internet and the open and shared experiment teaching mode thus effectively solving the problems of equipment shortages and lack of learning hours. It proposes equipment sharing, teacher sharing, course sharing and member sharing. During the experiment, students can share the development board with others through a reservation mechanism they can share the screen with others through remote control technology, and share the experiment process through real-time communication modules. Sharing allows students to learn the course content and complete experiments autonomously, freely, and without barriers. MOOE takes the sharing feature to the extreme, highlighting the sharing of people, ideas, and methods on the basis of resource sharing, completely changing the subordinate status of the students undertaking the experiment enabling them to enjoy resources and complete experiments at anytime, anywhere and on demand.

During the teaching process, we find that the hardware course is very practical. MOOE cannot replace the importance of classroom teaching. Therefore, we adopt a hybrid teaching model that combines "offline laboratory teaching and online MOOE teaching" to give full play to the advantages of both methods. According to actual measurement statistics, the hybrid teaching mode has obvious advantages when compared to the single offline teaching mode. The average number of questions answered by teachers increased by 7 times per person and the average tutoring time increased by 1.8 hours per person.

V. CONCLUSION

The "vertical" practice system that runs through software and hardware, innovates the practical teaching model, effectively breaking through the constraints of the "horizontal" practice systems that separates software from hardware on the development of comprehensive practice of systematic abilities training, and achieves the combination of intra- and extra-curricular and cross-curricular practice. The large-scale comprehensive practice project of the semester can be implemented step by step, creating a good experiment environment for students' which allows for independent experiments and personalized learning.

ACKNOWLEDGMENTS

This work was supported in part by “2030 Megaproject” - New Generation Artificial Intelligence (2020AAA0103503), Industry-University Cooperation Collaborative Education Projects held by the Ministry of Education and Baidu (152115PC01527).

REFERENCES

[1] Sahami M, Roach S, Cuadros-Vargas E, et al. ACM/IEEE-CS computer science curriculum 2013: reviewing the ironman report[C]//Proceeding of the 44th ACM technical symposium on Computer science education. 2013: 13-14.

[2] Coyle E J, Jamieson L H, Oakes W C. 2005 Bernard M. Gordon prize lecture*: integrating engineering education and community service: themes for the future of engineering education[J]. Journal of engineering education, 2006, 95(1): 7-11.

[3] Mao Y, Feng Y, Cheng D, et al. Computer curriculum system reform based on system ability training[C]//2016 11th International Conference on Computer Science & Education (ICCSE). IEEE, 2016: 907-910.

[4] De Michell G, Gupta R K. Hardware/software co-design[J]. Proceedings of the IEEE, 1997, 85(3): 349-365.

[5] Staunstrup J, Wolf W. Hardware/software co-design: principles and practice[M]. Springer Science & Business Media, 2013.

[6] Yan X, He F, Hou N, et al. An efficient particle swarm optimization for large-scale hardware/software co-design system[J]. International Journal of Cooperative Information Systems, 2018, 27(01): 1741001.

[7] Bencheva N, Kostadinov N, Ruseva Y. On Teaching Hardware/Software Co-design using FPGA[J]. Elektronika ir Elektrotechnika, 2010, 102(6): 91-94.

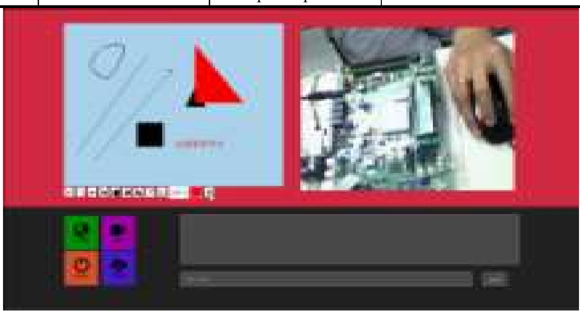
[8]Wang Z, Xu K, Wu S, et al. Sparse-YOLO: Hardware/Software co-design of an FPGA accelerator for YOLOv2[J]. IEEE Access, 2020, 8: 116569-116585.

[9] Li Y, Niu J, Zhang J, et al. MOOE: A new online education mode: Virtual simulation experiment MOOE platform for FPGA[C]//2016 IEEE Frontiers in Education Conference (FIE). IEEE, 2016: 1-8.

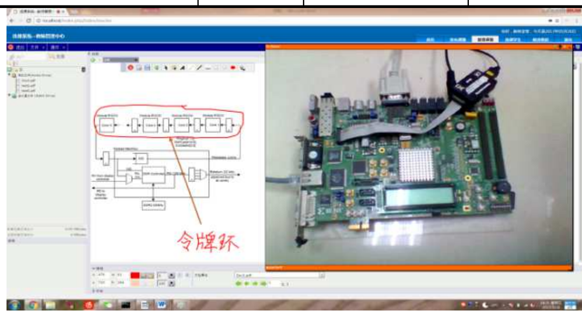
[10] Qiu J, Wang J, Yao S, et al. Going deeper with embedded fpga platform for convolutional neural network[C]//Proceedings of the 2016 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays. 2016: 26-35.

Table 1 Levels and Requirements of Experiments

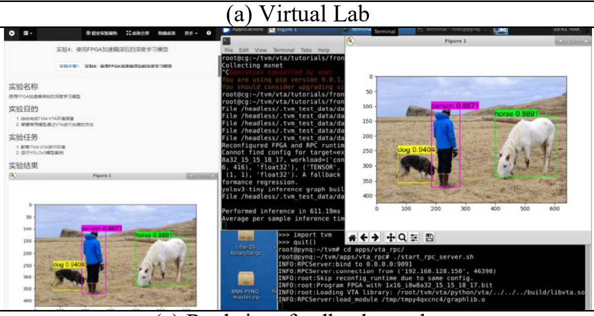
Level	Topics	Materials Provided	Contents	Objectives		
				Knowledge	Skill	Competency
Basic Experiment	Run the convolutional neural network (VGG) on an FPGA platform based on the TVM deep learning compiler	1) Provide "PC+ARM+FPGA" experimental development platform 2) Provide MNIST datasets 3) Provide TVM deep learning compiler 4) Provide ARM cross-compilation toolchain	1) Independently design lightweight VGG16 neural network;①Pruning: Delete the fully connected layer and bias items in the original VGG16 to reduce network parameters;② Simplification: Choose simple pooling function, activation function and loss function to reduce the computational complexity of FPGA;③ Quantization: Map the value of the network 2) Data processing:① Process the given MNIST data set and load the data correctly;② Split the data set into three parts: training, testing, and validating 3) Iterative training: obtain a neural network that meets the needs through iterative training 4) Compile the network: Use the TVM deep learning compiler to transform the trained VGG16 network down into an optimized hardware implementation; 5) Hardware kernel download: Download the hardware kernel compiled by TVM to FPGA through the ARM cross-compilation tool chain	Master the compression method of VGG network (pruning, simplification, quantization)	Master the execution process of "PC+ARM+FPGA" heterogeneous platform	Understand the design method and implementation principle of VGG convolutional neural network
Comprehensive Experiment	Design an ARM cross-compilation tool chain to realize the data communication between the upper layer PC and the bottom layer Fpga	5) Do NOT provide ARM cross-compilation toolchain 6) Provide FPGA dedicated convolution algorithm library	6) Design ARM communication program independently: in the prediction stage, for the trained lightweight VGG16 network, ARM uses I/O control methods (such as polling, interrupt, DMA, etc.) to call the FPGA dedicated convolution algorithm library to achieve data communication	Master the data interaction process of "PC+ARM+FPGA"	In combination with theoretical aspects be able to make qualitative and quantitative analysis on experiment process and results	Pay attention to the cultivation of engineering thinking and innovative thinking with basic hardware programming skills
Engineering Experiment	Design FPGA convolution algorithm library to accelerate VGG model operations	7) Do NOT provide FPGA dedicated convolution algorithm library	7) Independently design FPGA dedicated convolution algorithm library: use Verilog hardware description language to develop matrix convolution operation library, and provide standard calling interface	Master the Verilog method to achieve high-	Master the design and debugging methods of multi-core software and hardware and	Understand the difference between hardware programming and software
Research Experiment	Use Verilog hardware description language to develop convolutional neural network	8) Do NOT provide "PC+ARM+FPGA" experimental development platform	8) Independently build an FPGA-based software and hardware collaborative heterogeneous development platform 9) Independently design FPGA convolutional neural network	Master the method of Verilog to realize CNN	Research the method of accelerating convolutional neural network with FPGA	Have the ability to acquire and use relevant technical materials



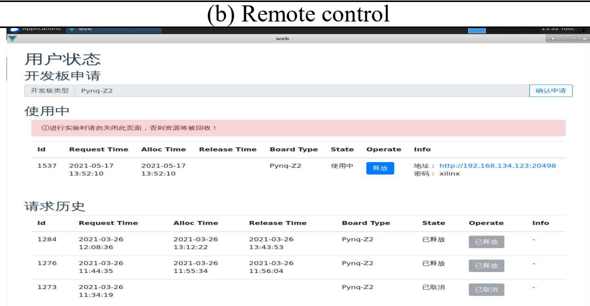
(a) Virtual Lab



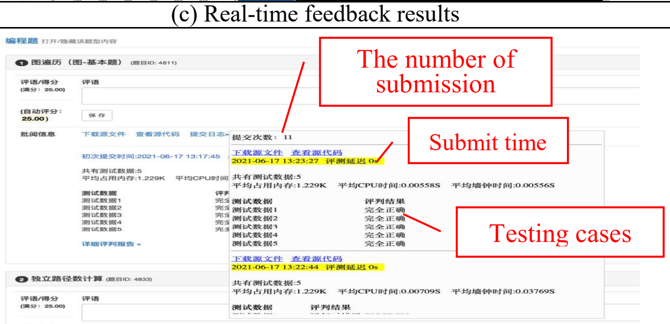
(b) Remote control



(c) Real-time feedback results



(d) Records of learning behavior data



(e) Follow the progress of the experiments

(f) Course Forum

Figure 4 MOOE Experiment Platform for Hardware-Software Co-design