Higher Education and Practice Vol. 3 No. 8, 2026

Academic Education Publishing House

Project-Based Teaching in a Deep Learning Course: Practice in a Data Science Program at an Application-Oriented University

Junli Zhang[1,*], Zimeng Wang[2]

[1]*Xi'an Eurasia University, Xi'an, Shaanxi, China*

[2]*Electronic Information Engineering, Xi'an University of Posts and Telecommunications, Xi'an, Shaanxi, China*

**Corresponding Author*

Abstract: Deep learning courses are conceptually abstract and involve a long engineering chain. Organized as lectures plus verification labs, they often leave students in an application-oriented data science program able to run code without being able to explain why a model works, and unable to transfer what they have practised to a new problem. This paper reports a project-based teaching practice in a 48-hour Deep Learning course. The content was reorganized into seven modules; instruction followed a path of problem-driven learning, knowledge construction and transfer; project tasks were arranged at four levels, from reproducing classic models to transfer, real application and final presentation; and five module assignments, one unit test and one project report carried the process assessment. A nine-step class procedure gave students about one third of the contact hours for self-study, group inquiry and coding. Over several rounds, project work was completed more often, topics moved from given datasets to real ones such as sheep breed recognition and leaf disease segmentation, and instructors spotted misunderstandings earlier by asking students to explain code on the spot. Limited contact hours, unclear individual contribution in group work and the assessment difficulty introduced by generative AI tools are also discussed.

Keywords: Deep Learning Course; Project-Based Teaching; Application-Oriented University; Process Assessment; Teaching Practice

1. Introduction

In the curriculum of data science and big data technology, Deep Learning usually comes after Machine Learning and is an application-oriented elective. The course deals with convolutional neural networks (CNNs), object detection and image segmentation, which are conceptually abstract, structurally complex, and sit at the end of a long engineering chain running from data preparation to model deployment. For an application-oriented university the difficulty lies not only in the knowledge itself: project-based teaching is repeatedly advocated because it narrows the gap between the classroom and real engineering practice [1]; and now that large models have entered everyday development environments, the objectives of the course have to move up as well, from writing code to solving complex problems and working with AI tools [2]. This places high demands on how the course is organised.

After several rounds of teaching the same situation kept recurring: students ran an image classification example step by step in the lab, yet could not explain in their reports what the convolution kernel actually does; they could reproduce the examples given in class but did not know where to start with another dataset. The final computer-based test usually produced a decent mark, but few students could carry a complete visual task through to the end in their graduation project. Part of the gap lies in the difficulty of the subject itself and part in the way the course is organised. The conventional arrangement cuts the course into three parts: theory, verification labs and a final examination. The knowledge points are coherent, while the students own questions remain fragmented. Project-based learning starts from a driving question and lets students construct knowledge while solving it. A recent systematic review in engineering education found that complex projects with driving questions promote engagement and deeper understanding [3,4]. In China, Liu Jingfu and Zhong Zhixian set out the

definition, elements and procedure of project-based learning (PBL) at an early stage [5], and reviews of active learning in engineering education provide more systematic evidence [6]. Project-based teaching is not easy to put into practice, however. If projects are confined to the end of the term they degenerate into one large assignment; if they are spread over the term they collide with the teaching schedule. This paper reports one concrete attempt: the course team divided project tasks into four levels by difficulty and embedded them in the 24 class sessions, and compressed the time given to continuous lecturing so as to increase the time for self-study, group inquiry and coding. The paper does not claim that this model is universally applicable; it sets out what was done, what difficulties were met and what changed over several rounds, for the reference of similar courses.

2. Overall Design of the Course and of Project-Based Teaching

2.1 Basic Information about the Course

Deep Learning is a specialised course in the data science and big data technology programme, worth 3 credits and 48 class hours, of which 32 are theory and 16 are practice. Research-oriented teaching has already been attempted in deep learning courses in China, using research cases and subject competitions to drive student development [7]; the author previously reformed a machine learning course along outcome-based education (OBE) lines [8], and that experience shows that project orientation works in an application-oriented university. It cannot, however, simply be transplanted into a deep learning course: the computational load is larger, the model structures are more complex, and the road from data to result is longer.

The content is organised into seven modules, and the allocation of hours is given in Table 1. Modules 1 to 3 deal with concepts and principles; Module 4, image classification, is the core and takes 14 hours; Modules 5 and 6, object detection and image segmentation, take 6 hours each; and the 4 hours of Module 7 are devoted entirely to project practice and the exchange of reports. The 24 sessions run at 2 to 4 hours per week, from the course overview in week 1 to the project report session in week 16.

Table 1. Course Modules and Allocation of Class Hours

Module	Main content	Theory hours	Practice hours	Total
Module 1 Introduction to deep learning	Computer vision and deep learning, application scenarios, framework and environment setup	2	0	2
Module 2 Basic image processing	Reading and writing images, colour spaces, geometric transforms, image enhancement	6	2	8
Module 3 Convolutional neural networks	Deep CNNs, principles and practice of convolution and pooling	8	0	8
Module 4 Image classification	LeNet-5, AlexNet, VGG, Inception, ResNet, DenseNet, MobileNet, transfer learning	8	6	14
Module 5 Object detection	Faster R-CNN, SSD, YOLOv3 and training detection networks	4	2	6
Module 6 Image segmentation	U-Net, Mask R-CNN and training segmentation networks	4	2	6
Module 7 Project practice based on deep learning	Writing project reports, presenting and discussing project work	0	4	4
Total	—	32	16	48

2.2 Tiered Design of the Project Tasks

The most easily overlooked issue in project-based teaching is the difficulty gradient. Tasks that are too easy are finished by following a tutorial; tasks that are too hard leave half the class stuck on environment configuration. The course team divided the projects into four levels, each of which makes new demands on the capabilities built at the previous level (see Table 2).

The point of the tiers is not to multiply projects but to give students of different backgrounds a place to start. The level of reproducing principles is about understanding: students should be able to explain, with intermediate feature maps and loss curves, what convolution, pooling, activation, loss and backpropagation

each do. The level of transfer and modification introduces change: the dataset is replaced, the network structure has to be altered, metrics have to be compared, and students must judge for themselves which option is more suitable. The level of real application moves the task into a scenario outside the course, where problems of annotation quality, class balance, missed detections and false positives come up one after another. The level of integrated presentation is about expression and reflection: students have to explain their work to classmates and answer the question of what they would change if they started again.

Table 2. Four Levels of Project Tasks and the Evidence Used for Assessment

Level	Related modules	Typical tasks	Assessment evidence
Reproduction of principles	Modules 2, 3, 4	Visualisation of convolution and pooling, LeNet-5 handwritten digit recognition, reproduction of five-class flower classification	Class worksheets, annotated code, in-class questioning
Transfer and modification	Module 4	Image classification on a different dataset, e.g. sheep breed recognition, road defect recognition	Lab logs, comparison table of metrics, analysis of error cases
Real application	Modules 4-6	Object detection or image segmentation in a scenario of the students' own choice, e.g. helmet detection, leaf disease segmentation	Data statement, model card, demonstration prototype, defence records
Integrated presentation	Module 7	Completion of the project report and presentation of the work	Project report, poster, peer assessment

2.3 How the Teaching Proceeds

The course proceeds in three steps: problem-driven learning, knowledge construction and transfer to new situations. Each module begins with a problem scenario. Before convolution is taught, students are shown a photograph on which edge detection has failed and asked to discuss what went wrong; before object detection, a video from an intersection is played and students count the vehicles that are missed. With the question posed first, the subsequent explanation has somewhere to land. Knowledge construction follows the order of the modules, but the way it is taught involves choices: derivations keep their main line, and more time is given to code and visualisation. Transfer and innovation fall mainly on project work and homework. For Session 13, for example, the required homework is to complete five-class flower classification with Inception V1, then extend the ResNet residual block to a two-layer stack and retrain for three-class flower classification, comparing the two on the same dataset. Such tasks have no standard answer, and students have to try for themselves.

3. Implementation: A Single Session as an Example

3.1 Design of the Session and Allocation of Time

Take the session on classic convolutional neural networks: Inception and ResNet. It is used here to illustrate implementation because it is a hinge: the two previous sessions have just covered AlexNet and VGG, and students have formed the impression that deeper is better, while Inception and ResNet correct that impression from the directions of width and depth respectively. The multi-scale parallel structure of Inception [9] and the residual learning of ResNet [10] are the hardest parts of the whole image classification module. The flower dataset for this session has five classes, with 952 training and 408 validation images, and the code environment is Jupyter Notebook with TensorFlow 2.x; students have to complete the Inception block and the ResNet residual block in class. The lesson plan follows the idea of teaching less and learning more, dividing the 90 minutes into nine phases (see Table 3).

Table 3. Teaching Phases and Time Allocation of Session 13

Phase and time	Teacher activity	Student activity
Contextual introduction (10 min)	Reviewing AlexNet and VGG; showing the curve in which the training error of a deeper network rises again; posing the core question	Observing the degradation phenomenon and offering their own conjectures
Self-study: Inception (10 min)	Distributing the worksheet and prompting students to note the four parallel paths and the role of the 1×1 convolution	Reading pages 60-72 of the courseware and completing Group A tasks independently

Group inquiry: Inception (10 min)	Circulating among six groups and giving hints on the dimensionality reduction achieved by the 1×1 convolution	Agreeing on a group answer and comparing parameter counts with and without the 1×1 convolution
Presentation: Inception (8 min)	Organising peer assessment and explaining the four-way parallel structure and the use of concatenate	One group presenting, the others adding comments and asking questions
Coding practice: Inception (10 min)	Providing the code skeleton of the Inception block and explaining how the channel numbers of each path are set	Completing the code in Jupyter and inspecting the model structure
Self-study: ResNet (8 min)	Prompting students to note the residual block and the shortcut connection	Completing Group B questions on the worksheet
Teacher exposition: ResNet (12 min)	Explaining the degradation problem, the residual idea and gradient backpropagation; comparing the design of the two networks	Taking notes on key formulas and structure diagrams and joining the discussion
Coding practice: ResNet (12 min)	Explaining the difference between add() and concatenate() and demonstrating three-class flower training	Completing the residual block function and watching validation accuracy
Summary and reflection (10 min)	Guiding students to draw a knowledge map and assigning the required and optional homework	Drawing the knowledge map and recording the requirements

It is worth accounting for the time carefully. Of the 90 minutes, continuous lecturing is held to about 30 minutes and the rest is handed to the students. The first time the session was run this way, phases two and three ran badly over time: two groups kept arguing during the presentation about what the 1×1 convolution does, and only eight minutes were left for ResNet. After the second adjustment, the role of the 1×1 convolution was made a compulsory item on the worksheet, so students had to write down their understanding during self-study, and the time for group inquiry fell accordingly. Submission of homework changed as well: the proportion submitting the required task (the flower classification comparison between Inception and ResNet) rose from about 70% to over 90%, and five or six students in each class voluntarily completed the optional task, an investigation of the convolution factorisation in Inception V3 with a calculation of how the parameter count changes.

3.2 Two Situations Observed in Class

One is visible understanding. In the coding phase students are asked to explain key lines on the spot, for example why the shortcut branch uses a 1×1 convolution to raise the number of channels to 256. Those who can answer also handle channel mismatch on their own later in real projects; those who cannot, and merely copy, get stuck on shape errors as soon as a real project begins. Such differences are invisible in a report but obvious once students are questioned in the lab. The other is worksheets completed as a formality. Some students treated the worksheet as a fill-in-the-blank exercise and copied half a sentence from the courseware for question A3. The course team therefore rewrote some questions as write it in your own words and give a counter-example, after which copying was of no use and students had to go back to the courseware and the code to think.

4. Assessment and Process Evidence

4.1 Assessment Structure

The course mark consists of continuous process assessment and a final computer-based test, 50% each. Within the continuous assessment, assignments account for 30%, practice for 10% and class performance for 10%, carried by five module assignments, one unit test on convolutional neural networks and one project report, assessed on a rolling basis as the modules proceed (see Table 4). This arrangement makes it difficult to pass by cramming at the end of term: every module has something that must be handed in.

Table 4. Carriers of Process Assessment and Their Requirements

Carrier	Related module	Task	Submission requirement
Assignment 1	Module 1	Propose a deep learning model that could solve a real problem, specifying its input, output, function and user interface	Design document

Assignment 2	Module 2	Convert a colour photograph into a pencil sketch using image algebra	Code (PDF) with line-by-line comments
Unit test	Module 3	Computation of convolution and pooling, output size and parameter count, classic network structures	Online test completed within two weeks
Assignment 3	Module 4	Train an image classification model and evaluate its performance	Code and project report
Assignment 4	Module 5	Train an object detection network and interpret the detection results	Code (PDF) with key comments
Assignment 5	Module 6	Train an image segmentation network and interpret the segmentation results	Code (PDF) with key comments
Project report	Module 7	Choose one of image classification, object detection and image segmentation and complete a full project report	Word and PDF versions of the report

4.2 How Process Evidence is Produced and Kept

Assignments 2 and 5 require the code to be submitted as a PDF with line-by-line comments, so students must state what each line does. Assignment 3 also requires a performance evaluation: reporting a single accuracy figure is not enough, and students must explain how the test set was split and whether class imbalance occurred. The project report must contain five parts, namely background, data description, data processing, model training and interpretation of results, of which interpretation is the section most often returned for revision. Many first drafts simply pasted the training curves, and the course team required an error analysis showing at least the circumstances in which three typical error cases occur. These materials do more than carry marks. They preserve the students learning process, so that later, when writing references, forming competition teams or assessing scholarships, evidence can be found in the records rather than resting on a single examination result.

5. Effects of Implementation and Discussion

The effects of the course are judged mainly from process materials: class worksheets, lab logs, project reports, posters and presentation records. These record what students actually did and how they did it, and they say more about the changes brought by project-based teaching than a single examination mark.

5.1 Three Changes Reflected in the Process Materials

5.1.1 Topics moved from given datasets to real scenarios

Taking the practice reports of the last two rounds as an example, a considerable share of the students working on image classification went beyond the datasets provided in the course, producing topics such as recognition of different sheep breeds (comparing ResNet-50 and MobileNetV2 on the same dataset), road defect recognition, and semantic segmentation of plant leaf diseases (with DeepLabv3-ResNet50 as the backbone and pixel accuracy and mIoU as the evaluation metrics). Such data have to be collected, cleaned and annotated by the students themselves, which is a level harder than working with a ready-made dataset.

5.1.2 Help-seeking shifted from asking first to checking first

Students used to raise a hand as soon as an error appeared; now they more often read the log and consult the documentation first, and bring the problem to class only when they cannot solve it. What is heard in the lab has changed from teacher, there is an error here to teacher, my validation accuracy is stuck at 60%, have I overdone the data augmentation?

5.1.3 Outcomes moved from a single report to demonstrable work

Beyond the project report, students began to produce posters and demonstration interfaces on their own initiative. One group built a simple upload interface for their object detection project, connected the trained model to it, and demonstrated recognition live during the defence by taking a photograph with a mobile phone. Work of this kind goes beyond the minimum requirements of the course, and for the students it is a first attempt to connect the course content with a real problem.

5.2 Problems that Remain

Limited contact hours are the most immediate problem. Seven modules have to be covered in 48 hours while project practice cannot be cut, and in some rounds the schedule was pushed beyond class time. The current solution is to move part of the coding practice to hours

outside class and hold the pace with worksheets, but this makes it easier for students who invest little outside class to fall behind.

Distinguishing individual contribution in group work remains unsatisfactory. Projects are mostly done in groups of three or four and the report is signed jointly, so individual input is hard to see in the result. The course team has tried peer assessment within groups and having each member defend a separate part, which improved discrimination but added considerably to the assessment workload.

The arrival of generative AI tools makes the question of how much of a report the student wrote a serious one. Students using a tool to explain an error or generate a code fragment is not in itself a violation, but without a trace of that use it is hard for the teacher to judge the student real level; the ethical risks have been discussed in detail [11], and studies have also tried to bring generative tools into project-based learning in laboratory teaching and to propose workable ways of organising the class [12]. The course team approach is as follows: the report keeps a statement specifying which parts were aided by a tool and what the student verified independently; explaining key code on the spot is a compulsory part of the defence, and any part the student cannot explain is not counted towards the individual mark. In practice this works better than simply banning the tools, and it still needs to be refined.

6. Conclusions and Next Steps

6.1 Some Lessons

In turning this course into what it is now, the most time-consuming part was not designing projects but aligning projects, class hours, assignments and assessment with one another. A few lessons: projects must be tiered, otherwise weaker students have nowhere to start and stronger ones find them useless; process evidence must land in concrete materials, since repeating in class that process matters means nothing; and lecturing time must genuinely be released, with the upper limit written into the lesson plan, or it creeps back during the term.

6.2 What We Plan to do Next

Three things are planned. First, to fix the task descriptions and assessment rubrics for the four levels and turn them into a version that a new teacher can use directly, so that the course depends less on one person. Second, to add quantitative records of learning input, code quality and defence performance, so that the evaluation of effects moves from the present qualitative description towards comparison across rounds. Third, to write the rules for using generative AI tools into the course documents, specifying which parts may be assisted by tools and which must be done independently, and to adjust the format of the defence accordingly.

Acknowledgments

This paper is supported by Shaanxi Province Education Science 14th Five-Year Plan 2024 project "Research on teaching reform of the machine learning course based on industry-education integration under the background of new engineering" (No. SGH24Y2632); Project of Computer Education Society (2026): “Exploration and Practice of Project-based Teaching in Deep Learning Courses” (No. SJJG-26-07).

References

[1] Zhong Q B, Shao Q J, Xie X H, et al. Construction of a project-based teaching model for engineering in local application-oriented universities: a case study of the Robot Institute of Ningbo University of Technology. Research in Higher Education of Engineering, 2025(3): 61-66+78. (in Chinese)

[2] Liu X F, Chen W, Han X, et al. Challenges and opportunities of experimental computer teaching in the era of large language models. Experimental Technology and Management, 2025, 42(8): 36-44. (in Chinese)

[3] Lavado-Anguera S, Velasco-Quintana P J, Terrón-López M J. Project-based learning (PBL) as an experiential pedagogical methodology in engineering education: a review of the literature. Education Sciences, 2024, 14(6): 617.

[4] Blumenfeld P C, Soloway E, Marx R W, et al. Motivating project-based learning: sustaining the doing, supporting the learning. Educational Psychologist, 1991, 26(3/4): 369-398.

[5] Liu J F, Zhong Z X. Research on the project-based learning (PBL) model. Studies in Foreign Education, 2002, 29(11): 18-22. (in Chinese)

[6] Prince M. Does active learning work? A

review of the research. Journal of Engineering Education, 2004, 93(3): 223-231.

[7] Ma Y K, Li X W, Gu L S, et al. Exploration of a research-oriented teaching model for the deep learning course. Journal of Higher Education, 2026(6): 129-132, 137. (in Chinese)

[8] Zhang J L. Exploration and practice of a machine learning course based on OBE and moral education. Journal of China Multimedia and Network Teaching, 2024(4). (in Chinese)

[9] Szegedy C, Liu W, Jia Y, et al. Going deeper with convolutions Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015: 1-9.

[10] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016: 770-778.

[11] Wu D, Wu H J. Potential risks of the educational application of general large models and how to avoid them: from the perspective of technological ethics. Journal of East China Normal University (Educational Sciences), 2024, 42(8): 64-75. (in Chinese)

[12] Ling J, Zhang L L, Chen H G, et al. Application of generative artificial intelligence in experimental computer teaching. Computer Education, 2025(5): 28-32. (in Chinese).