ISSN: 2637-6067 DOI: 10.6918/IJOSSER.202512_8(12).0007

Research on Computer Vision Teaching Reform Based on Productive Failure Cases

Junnan Hu, Busheng Li, Yan Zhao, Zihui Hu

School of Information Engineering, Jingdezhen Ceramic University, Jiangxi, Jingdezhen, CO 333403, China

Abstract

In response to issues such as the limited algorithm debugging proficiency and the disjunction between theory and practice in computer vision courses, this paper puts forward a teaching reform model grounded in the "Productive Failure" theory. Through the construction of a three - tier hierarchical case library encompassing code - level errors, algorithm logic deficiencies, and model tuning challenges, a reverse teaching path of "Failure Demonstration - Root Cause Analysis - Solution Iteration" is designed. This path is then integrated with a dynamic evaluation mechanism to monitor students' competence development.The case library consolidates high - frequency errors from student experiments. It combines open - source tools and online platforms to realize dynamic updates and hierarchical adaptation of cases. Practical implementation indicates that this model notably enhances students' engineering practice capabilities. Specifically, 84% of students can independently rectify code - level errors, with the average debugging efficiency increasing by 28%. In model tuning tasks, 32% of students proposed innovative solutions, among which 19 items were incorporated into the iterative case library.This reform validates the efficacy of "Controlled Failure" in cultivating technology transfer and innovation abilities, offering a replicable model for the practical teaching of computer vision courses.

Keywords

Computer Vision; Productive Failure; Hierarchical Case Library; Reverse Teaching; Dynamic Evaluation.

1. Introduction

In recent years, computer vision technology, as a core sub - field of artificial intelligence, has been extensively utilized in fundamental tasks such as image classification and object detection [1, 2]. Nevertheless, traditional teaching typically encounters the drawback of "emphasizing theory at the expense of debugging". Even though students are capable of replicating classic algorithms, they lack systematic analytical abilities for model failure scenarios [3]. Research demonstrates that approximately 73% of practical failures are attributed to environment configuration errors and parameter logic defects. Meanwhile, only 12% of the existing teaching case libraries encompass systematic root - cause analysis and repair training for failure scenarios, which underscores the necessity of teaching reform [4].

Contemporary reforms in computer vision courses primarily concentrate on the innovation of teaching modes (e.g., blended online - offline teaching [5, 6]) and the enhancement of practical abilities (e.g., project - driven learning [7]). Nevertheless, the existing explorations are still confronted with two significant limitations. Firstly, the teaching content is predominantly "success - oriented," lacking in - depth analyses of typical failure scenarios. This deficiency makes it arduous for students to develop problem - solving thinking. Secondly, the evaluation

ISSN: 2637-6067 DOI: 10.6918/IJOSSER.202512_8(12).0007

system overly depends on outcome indicators (such as model accuracy), neglecting the cultivation of cognitive iteration and innovation capabilities during the debugging process [4].

In response to these issues, the "Productive Failure" theory presents a novel perspective for resolution. By pre - setting hierarchical failure scenarios and guiding students through the cognitive cycle of "error attribution - solution reconstruction - knowledge transfer" [8], this model has been demonstrated to enhance debugging efficiency by over 25% in programming education [9].

This paper proposes a computer vision teaching framework based on the Productive Failure theory. Through constructing a three-tier hierarchical case library (code-level errors, algorithmic logic defects, model tuning challenges), designing a reverse teaching path, and combining it with a dynamic evaluation and feedback mechanism to track competency growth. The case library integrates high-frequency errors from student experiment logs, relying on the EduCoder platform for dynamic adaptation of hierarchical cases. Pilot data show that the average student debugging time decreased from 4.2 hours to 2.8 hours, and 86% of participants could independently locate the root cause of complex model failures; furthermore, 19 student optimization schemes were incorporated into the iterative case library, verifying the effectiveness of the "failure-innovation" transformation mechanism. This paper provides universal reference for practical teaching reform in computer vision courses.

2. Hierarchical Failure Case Library

To systematically resolve the core contradiction in computer vision courses where "theoretical understanding is adequate but practical debugging is difficult", this paper uses image classification tasks as the carrier to construct a three-tier hierarchical case library covering the code level, algorithm layer, and model optimization layer. The case library integrates high-frequency error scenarios from student experiment logs, generating teaching resources through standardized processes like "error collection - categorization and attribution - teaching adaptation". It designs reproducible failure situations based on classic datasets like MNIST and CIFAR-10, combined with Jupyter Notebook debugging plugins and OpenCV visualization tools, building a gradient training system of "basic error fixing→logical defect diagnosis→tuning strategy innovation". The dynamic update rate of the case library is ≥40%, cumulatively collecting 142 cases, covering 93% of typical error types encountered in student experiments.

2.1. Code-Level Error Scenarios

Code-level errors represent the primary impediment in practical applications, predominantly concentrated in the stages of environment configuration and data processing. To tackle Python environment dependency conflicts, the case library simulates image reading exceptions stemming from OpenCV version incompatibilities (e.g., cv2.imread returning None), guiding students to rectify environment configurations by comparing the parameter discrepancies of library functions across different versions. At the data preprocessing stage, it devises dataset loading failure cases resulting from path encoding errors (the absence of backslash escape characters in Windows systems), necessitating students to identify the origin of the FileNotFoundError through log analysis. A typical instance is the activation value distribution shift due to the omission of normalization in the MNIST dataset: when transforms.Normalize is not employed, the outputs of the Sigmoid layer converge within the range of 0.48 - 0.52, leading to a 22% decline in model accuracy.

During the teaching process, instructors embed erroneous code into Jupyter Notebooks and leverage ipywidgets to develop interactive debugging plugins. When a RuntimeError is triggered, guidelines for tensor dimension correction (such as instructions for the unsqueeze(0)

ISSN: 2637-6067 DOI: 10.6918/IJOSSER.202512_8(12).0007

operation) are automatically pushed, enabling 84% of students to independently resolve single-image prediction dimension mismatch issues within 15 minutes.

2.2. Algorithm Logic Defects

Algorithm - layer errors primarily pertain to logical fallacies and parameter misapplications during model construction. In the design of convolutional neural networks, the case library compiles instances where the misplacement of fully - connected layers disrupts feature maps. For example, students may erroneously position nn.Linear layers prior to convolutional layers, resulting in a 32×32 input image being inappropriately flattened into a 3072 - dimensional vector. This phenomenon significantly degrades the model's accuracy to the level of random guessing (10% in the CIFAR - 10 task). By leveraging PyTorch hook functions to capture the output shapes of each layer, students can visually observe the dimension collapse process, thus comprehending the sequential logic of network structures.

Another typical problem is the mismatch between loss functions and output layers. For instance, directly invoking CrossEntropyLoss without incorporating LogSoftmax can lead to abnormal gradient calculations. In terms of teaching strategies, TensorBoard is employed to visualize the loss oscillation curves during training. Comparative experimental groups (with correct and incorrect loss function configurations) are established, enabling students to understand the internal algorithmic relationships through parameter sensitivity analysis.

Moreover, regarding the channel order errors in HSV color space conversion (misassigning the Hue channel to the B channel), the case library offers OpenCV dynamic visualization tools. These tools can display in real - time the impact of hue distortion on classification models, thereby enhancing students' understanding of feature engineering logic.

2.3. Model Tuning Challenges

Model optimization layer cases are designed to train students to address complex issues such as overfitting and convergence difficulties. To tackle the problem of poor generalization due to data distribution shift, the case library constructs a scenario with a lack of data augmentation in the CIFAR - 10 dataset. Specifically, the training set consists solely of standard images with white backgrounds, while the test set incorporates random noise backgrounds, leading to a validation accuracy gap of up to 36 percentage points. Students are required to devise augmentation strategies, such as RandomRotation and ColorJitter, to enhance the robustness of the model.

In the context of optimizer configuration, by simulating the gradient explosion phenomenon triggered by excessively high learning rates (for instance, when the Adam optimizer uses the default learning rate of 0.001, the loss value soars to NaN), students are led to comprehend the critical thresholds for parameter adjustment through comparing the slopes of loss curves under different learning rates.

Regarding deep network training, the case library designs scenarios involving gradient vanishing problems caused by the absence of Batch Normalization (BN) layers. For example, an 8 - layer CNN without BN layers reaches a stagnant validation accuracy of 68% on the MNIST task, which increases to 96% after the addition of BN layers. During the teaching process, the gradient heatmap analysis tool in the Captum library assists students in identifying gradient decay areas within network layers, enabling them to autonomously design BN layer insertion schemes.

Through these challenging tasks, 32% of students proposed innovative optimization methods. Among them, a hybrid augmentation strategy based on CutMix reduced the overfitting rate by 19% and was included in the case library as advanced teaching material.

ISSN: 2637-6067 DOI: 10.6918/IJOSSER.202512_8(12).0007

3. Reverse Teaching Path

This paper employs the reverse teaching approach of "Failure Demonstration - Root Cause Analysis - Solution Iteration," with typical failure cases serving as the entry points to facilitate students' cognitive reconstruction. In the failure demonstration stage, problem awareness is enhanced through the intuitive comparison of incorrect and correct results. For instance, in a YOLO object detection experiment, the missed detection of vehicles caused by inappropriate anchor box size settings is presented. The abnormal Intersection over Union (IoU) distributions between predicted boxes and ground - truth boxes are visualized using tools such as TensorBoard, and this is supplemented by heatmaps of target detection results (e.g., regions of undetected vehicles showing low - confidence distributions), thereby intuitively revealing the model's failure phenomena.

In the root cause analysis stage, hierarchical diagnostic strategies are employed. At the code level, the automatic syntax - checking feature of the EduCoder platform is utilized. At the algorithm level, classification preferences are analyzed through confusion matrices (e.g., class imbalance in ceramic defect detection). At the model tuning level, TensorBoard is used to visualize the anomalous fluctuations in training loss curves (e.g., oscillations resulting from improper learning rate settings).

In the solution iteration stage, gradient task chains are designed. Basic tasks involve rectifying code errors (e.g., adjusting the HSV channel order). Advanced tasks guide parameter tuning (e.g., optimizing YOLO anchor sizes). Challenge tasks encourage cross - scenario migration (e.g., applying threshold - processing experience to medical image segmentation).

Teaching practice indicates that 86% of students were able to independently complete basic debugging tasks, and 32% of the participants proposed innovative optimization schemes (e.g., improvements in data augmentation strategies), which verifies that the reverse teaching approach promotes students' problem - solving abilities.

4. Dynamic Evaluation and Feedback Mechanism

To quantify teaching efficacy, a dynamic feedback system integrating "process evaluation + competency growth tracking" was established. Evaluation dimensions encompass debugging efficiency (the time taken to initially locate an error), solution effectiveness (the improvement rate of model accuracy), and technical transfer ability (the cross - task code reuse rate).

During the implementation process, full - process tracking depends on information tools. The EduCoder platform automatically records experimental operation paths (such as the number of code modifications and debugging duration), and the Chaoxing Smart Teaching Platform collects pre - class completion rates and post - class project submission rates. These data, combined with teacher - annotated failure analysis reports, form multi - dimensional data support.

The feedback mechanism is implemented in three stages. Real - time feedback prompts common error patterns through intelligent pop - ups. Periodic feedback is based on teachers' personalized guidance on laboratory reports. Summary feedback utilizes end - of - semester "Error - Competency" mapping diagrams to visualize students' growth trajectories in aspects such as code debugging and algorithm comprehension.

Pilot results demonstrate that students' average debugging efficiency increased by 28% (from 4.2 hours to 2.8 hours), and 72% of optimization solutions were reproducible (for example, the HSV channel correction strategy was incorporated into the iterative case library). This proves that the dynamic evaluation mechanism effectively promoted technological transfer and the cultivation of innovation capabilities. This system breaks through the limitations of traditional

ISSN: 2637-6067 DOI: 10.6918/IJOSSER.202512_8(12).0007

accuracy - oriented evaluation, achieving the symbiotic evolution of teaching resources and learning outcomes.

5. Implementation Outcomes and Reflection

Following the implementation of the computer vision teaching reform grounded in productive failure cases, remarkable outcomes have been attained across diverse aspects. In terms of the enhancement of student competencies, there have been qualitative advancements not merely in specific skills such as code debugging and model tuning but also in problem - solving thinking and innovation capabilities, thereby exhibiting a revitalized vigor. The fact that 84% of students can independently rectify code - level errors implies that they possess robust self - resolution capabilities when confronted with fundamental coding problems and no longer overly depend on instructor guidance. An average improvement of 28% in debugging efficiency indicates that students can handle practical problems more effectively, identify and resolve issues more rapidly, and offer substantial support for the rapid iteration and optimization of their actual projects. In model tuning tasks, 32% of students put forward innovative solutions, among which 19 items were incorporated into the iterative case library, fully demonstrating that students can break away from conventional thinking patterns when dealing with complex model problems and propose novel solutions with a certain degree of practicality and generalizability.

From the perspective of optimizing teaching methods and resources, the establishment of the hierarchical failure case library offers abundant and targeted materials for teaching. By integrating high - frequency errors from student experiments and combining them with open - source tools and online platforms to achieve dynamic updates and hierarchical adaptation of cases, it is ensured that the teaching content can keep abreast of technological development and students' actual requirements. The generation of teaching resources through standardized processes such as "error collection - categorization and attribution - teaching adaptation" guarantees the quality of cases and the systematic nature of teaching. Designing reproducible failure scenarios based on classic datasets like MNIST and CIFAR - 10, in combination with Jupyter Notebook debugging plugins and OpenCV visualization tools, constructs a gradient training system of "basic error fixing → logical defect diagnosis → tuning strategy innovation", providing students with a progressive learning path from the simple to the profound.

Nevertheless, several issues that demand reflection emerged during the implementation of the reform. On one hand, despite the case library's dynamic update rate being ≥40%, with the rapid advancement of technology and the continuous evolution of student needs, the speed of case updates may still not fully meet the practical teaching requirements. How to further enhance the efficiency and quality of case updates, ensuring their timeliness and relevance, calls for further deliberation. On the other hand, although the dynamic evaluation and feedback mechanism realizes full - process tracking and multi - dimensional data support, there is still room for improvement in terms of the timeliness and personalization of feedback. Real - time feedback via smart pop - up prompts for common error patterns may not meet students' deeper needs in complex problems; periodic feedback based on teachers' personalized guidance on lab reports is restricted by teachers' energy and time, potentially preventing a detailed analysis of each student's issues. Therefore, optimizing the feedback mechanism to enhance its immediacy and degree of personalization is a crucial issue that needs to be resolved in subsequent reforms.

Furthermore, although the reform verified the effectiveness of "controlled failure" in cultivating technological transfer and innovation capabilities, how to better promote this teaching model to other computer - related courses and realize the scale effect of teaching reform is also a direction worthy of in - depth research. Simultaneously, during the process of teaching reform, balancing the relationship between "productive failure" teaching and traditional teaching, fully

ISSN: 2637-6067 DOI: 10.6918/IJOSSER.202512_8(12).0007

leveraging the advantages of both, and avoiding potential issues such as disjointed teaching transitions or student adaptation difficulties also requires further exploration and practice.

Acknowledgment

This work was supported in part by the Jingdezhen Ceramic University Degree and Graduate Education Teaching Reform Research Project titled “Focusing on ‘Digital Intelligence’: Innovative and Practical Graduate Teaching Model Integration of Industry and Education — A Case Study of the Big Data Science and Application Program.”

References

[1] Deshpande A R. Learning multiple solutions to computer vision problems [Z] (2020–05–05).

[2] Hsieh T-Y, Cheng C-C, Chao W-J, et al. On Development of Reliable Machine Learning Systems Based on Machine Error Tolerance of Input Images [J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2023, 42(4): 1323–1335.

[3] Jeon I-S, Kang S J, Kang S-J. A Staged Framework for Computer Vision Education: Integrating AI, Data Science, and Computational Thinking [J]. Applied Sciences, 2024, 14(21): 9792.

[4] Jeon I-S, Kang S J, Kang S-J. A Staged Framework for Computer Vision Education: Integrating AI, Data Science, and Computational Thinking [J]. Applied Sciences, 2024, 14(21): 9792.

[5] Cui Y, Ma Z, Wang L, et al. A survey on big Data-enabled innovative online education systems during the COVID-19 Pandemic [J]. Journal of Innovation & Knowledge, 2023, 8(1): 100295.

[6] Fedajev A, Jovanović D, Janković-Perić M, et al. Exploring the Nexus of Distance Learning Satisfaction: Perspectives from Accounting Students in Serbian Public Universities During the Pandemic [J]. Journal of the Knowledge Economy, 2024, 16(1): 4465–4495.

[7] Hassner T, Bayaz I. Teaching Computer Vision [J]. ACM Transactions on Computing Education, 2015, 14(4): 1–17.

[8] Quinn F, Hobbs L. “I’m on My Own and I’m Not Trained”: A Cultural-Historical Activity Theory Analysis of Teaching Mathematics Out-of-Field in a Small School [J]. International Journal of Science and Mathematics Education, 2024, 23(1): 1–23.

[9] Wang Y (Arthur). Unpacking the rhetoric of Diversity, Equity, and Inclusion Statements for academic job application purposes: A step-driven rhetorical move Study [J]. English for Specific Purposes, 2024, 75: 49–65.