

人工智能赋能下程序设计类课程教学模式创新

徐 琴¹, 乔宇龙², 吴梓杏², 张 迁^{2,*}

1. 金陵科技学院, 网络安全学院, 江苏 南京, 211169

2. 江苏开放大学, 信息工程学院, 江苏 南京, 210036

摘要: 本研究探讨了人工智能(AI)技术在程序设计课程教学中的应用及其创新路径。随着生成式AI的快速发展, 程序设计教育面临深刻变革。通过对现有研究的综合分析, 发现AI技术可有效赋能程序设计教学的多个维度, 包括教学内容更新、教学方法创新和学习评价优化。研究表明, 将AI融入程序设计教学可提高学生的学习动机和学习效果, 促进高阶思维能力培养。基于“人机协作”理念, 本研究提出了AI赋能程序设计教学的整合框架, 涵盖课前预习、课堂互动、课后巩固和项目实践四个环节, 并强调了任务驱动和个性化学习的重要性。同时, 针对可能出现的技术依赖与思维能力弱化风险, 提出相应的防控措施和平衡策略, 确保学生的综合能力和基础技能得到充分发展。本研究为程序设计课程教学改革提供了理论依据和实践指导, 对培养适应智能时代的创新型编程人才具有重要意义。

关键词: 人工智能; 程序设计类课程; 教学改革; 课程创新

Teaching Mode Innovation of Programming Courses Empowered by Artificial Intelligence

XU Qin¹, QIAO Yu-Long², WU Zi-Xing², ZHANG Qian^{2,*}

1. School of Network Security, Jinling Institute of Technology, Nanjing, Jiangsu 211169, China

2. School of Information Technology, Jiangsu Open University, Nanjing, Jiangsu 210036, China

Abstract: This study explores the application of artificial intelligence (AI) technology in the teaching of programming courses and its innovative path. With the rapid development of generative AI, programming education faces profound changes. Through a comprehensive analysis of existing studies, it is found that AI technology can effectively empower multiple dimensions of programming teaching, including the updating of teaching content, the innovation of teaching methods and the optimization of learning evaluation. Research shows that integrating AI into programming teaching can improve students' motivation and learning effect, and promote the cultivation of higher-order thinking ability. Based on the concept of "human-computer collaboration", this study proposes an integrated framework for AI-enabled programming teaching, which covers the four links of pre-study, classroom interaction, post-course consolidation and project practice, and emphasizes the importance of task-driven and personalized learning. At the same time, corresponding preventive and control measures and balancing strategies are proposed to address the possible risks of technological dependence and weakening of thinking ability, to ensure that students' comprehensive ability and basic skills are fully developed. This study provides theoretical basis and practical guidance for the teaching reform of programming courses, which is of great significance for cultivating innovative programming talents adapted to the intelligent era.

Keywords: Artificial intelligence; Programming courses; Teaching reform; Curriculum innovation

近年来, 生成式人工智能(AI)技术的迅猛发展正深刻改变着程序设计课程的教学模式、内容与方法。作为计算机科学教育的核心组成, 程序设计课程亟须适应AI时代的新需求, 培养具备创新能力和解决复杂问题能力的人才。

已有研究表明, AI技术在程序设计教学中具有重要价值。牟玉亭等人结合学习金字塔理论与AI大模型创新教学方法, 提高了学生学习主动性和效果^[1], 但其研究侧重于理论构建, 缺乏实证数

据支持。相比之下,穆肃等人提出“生成式 AI 促进分析持续生成”的技术指向^[2],其框架虽具可操作性,但未充分考虑学科特性差异,在程序设计领域应用时存在适配性问题。裴榕^[3]的人机协同观点为本研究提供了理论基础,但其论述缺乏微观教学环节的具体策略。

在课程融合方面,颜丽蓉等人^[4]与沈苑和汪琮^[5]的研究分别从课程设计和教学决策角度提出了 AI 融入路径,但二者均未系统解决程序设计教学中逻辑思维与技术应用的平衡问题。张玉柳和罗江华^[6]提出的“赋能学科教学适应性”概念有启发性,但其实践路径在编程教育中的适用性有待验证。

程序设计教育创新研究呈现多元化趋势,袁金凤基于“三链融合”的人才培养体系^[7]虽然创新性强,但未充分考虑 AI 工具的介入;李锋等人^[8]的项目活动模式与丁美荣和王同聚^[9]的三位一体模型虽各具特色,但均未将 AI 作为核心要素整合。任浩等人^[10]和王明芳^[11]的教学方法研究为本研究提供了方法论参考,但在 AI 环境下需要重新审视其有效性。

AI 与程序设计课程的深度融合研究方面,张彦芳等人^[12]的项目教学模式和苏小红等人^[13]的“四位一体”创新路线均试图系统整合 AI 与编程教学,但前者过于依赖特定项目,后者则缺乏对核心编程能力培养的具体策略。

现有研究虽为 AI 与程序设计教学融合提供了多维视角,但尚未形成系统性、全面性的教学模式和方法论体系,特别是缺少在 AI 如何真正赋能课程各环节、如何平衡 AI 辅助与基础能力培养等方面的探讨。本研究将立足 AI 赋能视角,融合现有研究的优势,构建一个更为完整的、适应 AI 时代特征的程序设计课程教学新范式。

1 人工智能赋能下程序设计类课程面临的挑战

AI 技术的迅猛发展正深刻改变着高等教育领域,程序设计类课程作为计算机科学教育的核心组成部分,面临着前所未有的挑战与机遇。随着大语言模型等生成式 AI 技术的普及,传统编程教学中的许多范式正被颠覆,从知识传授到能力培养,从标准化教学到个性化学习,从封闭环境到开放生态,教学的各个环节都需要重新审视与设计。

这种变革不仅仅是技术工具的更新换代,更是对教育本质的深层次思考。程序设计课程不再只是传授编程语言和算法,而是需要培养学生在 AI 辅助环境下的问题解决能力、创新思维与协作精神。教师们也需要从知识权威转变为学习引导者,重新定位自己在教学过程中的角色。面对这一系列挑战,传统教学模式与 AI 时代需求之间的差距日益凸显。

表 1 清晰展示了 AI 时代程序设计课程面临的七大维度挑战。在这种背景下,教育者需要重新思考课程的核心目标,培养学生不仅掌握编程语法,更要具备解决复杂问题的能力和与 AI 协作的技能。课程内容需要建立灵活的更新机制,以跟上技术的快速迭代;教学方法需要从传统讲授转向项目驱动、协作学习等互动模式;教师需要转变为学习引导者,掌握设计个性化学习路径的能力;评价体系需要从结果导向转向过程导向,引入多元评价方法;学习环境则需要整合线上线下资源,创造更加开放和动态的学习生态。

这些挑战虽然艰巨,但也蕴含着教育创新的机遇。通过系统性思考和创新实践,程序设计类课程可以更好地适应 AI 时代的需求,培养出具备未来竞争力的复合型人才。如何在保持教学本质的同时,有效利用 AI 技术增强教学效果,是每一位教育工作者需要思考和探索的重要课题。

表 1 传统教学模式与 AI 时代需求的差距

Table 1 The gap between traditional teaching models and the needs of the AI era

维度	传统教学模式	AI 时代需求	面临的挑战
教学理念	知识传授为中心	能力培养为中心	如何转变教学理念和目标
教学内容	静态、标准化	动态、个性化	如何更新课程内容以跟上技术发展
教学方法	讲授为主、互动有限	多元化、高互动性	如何创新教学方法以提高学习效果
教师角色	知识传授者	学习引导者	如何转变教师角色和提升相关能力
学生角色	被动接受者	主动学习者	如何激发学生的主动性和创造力
评价体系	结果导向、标准化	过程导向、多元化	如何建立更科学的评价机制
学习环境	封闭、静态	开放、动态	如何构建适应 AI 时代的学习生态

2 人工智能赋能的程序设计课程教学模式

2.1 AI 辅助教学设计与个性化学习路径

AI 辅助教学设计基于学习分析和教育数据挖掘，为程序设计课程提供智能化教学规划，将传统静态课程计划转变为动态适应的学习体系。这种模式通过采集和分析学生的编程习惯、错误类型、解题思路和学习进度等数据，构建学生的知识状态模型，推断其认知水平和学习需求，从而为每位学生定制个性化学习方案。

传统程序设计课程常依赖教师经验和预设教学目标，采用统一进度和内容，难以满足学生个体差异。而 AI 辅助教学设计能精准识别知识点之间的依赖关系和学习路径关键节点，自动调整教学内容顺序和难度梯度。例如，系统可发现学生在循环结构上存在薄弱，自动推荐相关练习；或识别学生已掌握基础语法，直接推送更高阶任务。

个性化学习路径实现方式主要包括智能预测学习瓶颈、自适应学习任务、多元化资源匹配和学习行为引导（见表 2）。通过这些技术手段，系统可以为不同学生提供差异化学习体验，更好地满足个体需求。基于学生的能力、学习风格和兴趣，AI 构建的个性化路径可呈现为项目选择、资源推荐、练习难度及评估形式的差异化。例如，对游戏开发感兴趣的学生可优先推荐小游戏开发项目学习数据结构；而对数据分析感兴趣的学生则可通过处理真实数据集掌握相同知识点。

表 2 个性化学习路径的实现方式

Table 2 A consideration of methodologies for the personalization of learning pathways

实现方式	技术手段	应用场景	实施难点
智能预测学习瓶颈	机器学习算法、知识图谱	编程技能诊断、难点预警	预测模型的准确性、数据收集
自适应学习任务	推荐系统、动态难度调整	编程练习、项目任务	任务库建设、难度标定
多元化资源匹配	内容标签化、相似度计算	补充学习材料、实例推荐	资源质量控制、标签体系构建
学习行为引导与干预	学习行为分析、智能提醒	编程习惯培养、时间管理	平衡干预与自主性

个性化学习路径实施依赖丰富的资源库和弹性课程结构。资源库需涵盖多层次、多类型学习材料和任务，以满足不同学习需求。教师角色需由知识传授者转为学习环境设计者和 AI 系统监督者。教师借助 AI 系统提供的数据分析与推荐，优化教学策略，在关键环节进行针对性指导。AI 辅助教学设计并非替代教师决策，而是增强教师教学支持和效率，使教师将更多精力投入创造性教学与深入辅导。

2.2 智能评估和反馈系统

智能评估系统在程序设计课程中不仅关注代码的正确性，还包括代码质量、算法效率和编程风格等多维度指标。这种系统通过自动化测试、静态代码分析、动态执行分析、代码相似度比较和问题解决过程追踪等技术，对学生代码进行全面分析和评价。智能评估的主要特点包括多维度评估、实时反馈、个性化分析、过程追踪和自动化处理。这种评估方法能够高效处理大规模班级的代码评估需求，为学生提供及时、全面的学习反馈。

智能反馈系统基于评估结果，为学生提供及时、个性化、具体的改进建议。其主要优势包括即时性、针对性、全面性、可视化和持续跟踪。学生可在编码过程中获得实时反馈，系统根据学生具体错误和历史表现提供个性化建议，不仅指出问题，还提供解决方案和学习资源。通过图表等方式直观展示代码质量和性能，同时记录学生进步情况，支持长期能力发展分析。

智能评估和反馈系统的应用显著提升了编程教学效果。它缩短了学习反馈循环，提高了学生自主学习能力，同时为教师提供了丰富的教学数据支持。然而，在应用过程中也需注意保持适度的人工干预，确保评估的公平性和全面性。表 3 对比了传统反馈与智能反馈的特点，凸显了智能反馈系统在编程教学中的优势。

表 3 传统反馈与智能反馈的对比
Table 3 Traditional feedback vs. intelligent feedback

特征	传统反馈	智能反馈
及时性	延迟，通常需要数天	即时，可在提交后立即获得
个性化	有限，通常为统一标准	高度个性化，基于学生表现
全面性	主要关注结果正确性	涵盖多个维度，包括过程分析
一致性	可能因评分者而异	标准一致，减少主观因素
可扩展性	受限于教师时间和精力	可轻松应对大规模班级

2.3 虚拟编程环境与仿真平台

虚拟编程环境与仿真平台是实现 AI 辅助教学设计和个性化学习路径的关键工具，同时为智能评估和反馈系统提供了实施基础。这些 AI 赋能的编程环境不仅包含传统 IDE 的功能，还融入了智能辅助、场景模拟和学习管理等先进特性，为程序设计课程提供了全方位的学习平台。

虚拟编程环境通过智能代码补全、实时错误检测和可视化调试等功能，有效支持个性化学习。基于学生知识状态模型，系统能提供针对性的代码建议和错误提示。这种即时反馈机制与智能评估系统紧密结合，使学生在编码过程中不断获得改进建议，加速学习循环。

仿真平台作为虚拟编程环境的延伸，构建了逼真的应用场景，将抽象的编程概念与实际问题解决紧密结合。它为学生提供了安全、可控且高度互动的学习环境，使他们能在接近真实世界的情境中应用和测试编程技能。仿真平台的主要特点包括：

- (1) 真实性与交互性：模拟各种实际工作环境，学生可实时交互并观察代码执行效果。
- (2) 可控性与安全性：教师可设计特定场景和任务，学生在隔离环境中安全实验。
- (3) 多样性与适应性：支持多种编程语言和框架，适应不同技术环境。
- (4) 协作与团队学习：支持多人协作，培养团队编程能力。
- (5) 性能分析与优化：提供代码效率和资源使用分析工具，助力程序优化。

这些平台的应用丰富了程序设计课程的教学手段。学生可以尝试复杂的编程任务，如构建分布式系统、开发移动应用或设计数据库架构，而无需搭建复杂的实际环境。这不仅节省了时间和资源，还让学生能够专注于核心的编程概念和问题解决策略。

3 创新教学方法的实践与应用

3.1 项目驱动与协作式编程

项目驱动学习通过具体项目的设计，将编程知识融入真实应用场景中，使学生在解决实际问题的过程中锻炼系统设计与代码实现能力。结合 AI 辅助，教学中可设计难易渐进的项目序列，将虚拟编程环境与仿真平台应用于项目实践，模拟电商平台、智能推荐系统等复杂场景，大幅增强学生的学习体验与应用意识。伴随敏捷开发理念的引入，学生能够通过需求分析、设计、编码、测试及评审等迭代流程，培养工程化思维和高效协作能力。同时，AI 辅助的项目管理工具能够帮助学生规划进度、跟踪任务，提升项目执行的规范性与透明度。

团队协作作为项目驱动的重要组成部分，贯穿结对编程和分布式远程协作等多种模式。结对编程注重双人互动，AI 系统则可作为“第三观察者”实时分析代码、给出改进建议，辅助解决技术难题；借助云端代码托管与协作平台，学生分布式协作能力得到锻炼，AI 辅助分支管理和冲突解决促进团队协作畅通无阻。通过角色轮换机制，学生能综合理解前端、后端、测试、架构设计等多种开发职责，AI 系统则根据个体能力推荐角色扮演及相应学习资源，确保每位成员多面发展。

表 4 总结了项目驱动与协作式编程的关键要素及 AI 技术的辅助功能，为教学设计和实施提供参考。AI 工具在此类教学中贯穿项目设计、团队管理、代码质量控制和过程评估全过程，成为提升教学深度与个性化的重要引擎。

表 4 项目驱动与协作式编程实施关键要素
Table 4 Key elements of project-driven and collaborative programming implementation

关键要素	内容描述	AI 辅助支持
项目设计	梯度化项目序列，真实场景模拟	项目难度匹配，自动生成需求文档
团队角色分配	前端、后端、测试等角色设置，定期轮换	基于学生能力的角色推荐
协作机制	结对编程，分布式协作，代码审查	代码质量实时分析，冲突解决建议
评估与反馈	过程性评估，代码质量评估	自动代码评测，贡献度分析

3.2 人机协作的混合教学模式

人机协作的混合教学模式将专业引导与智能服务有机结合。教学设计者制定教学目标和框架，重点培养学生的创造性思维与批判意识，AI 系统则承担个性化资源推送、常见问题答疑、学习数据分析等辅助职责。在课前，AI 为学生定制预习内容，课堂上讲解核心知识并与 AI 共同支持学生互动，课后 AI 提供个性化练习并反馈学习成效。不同的混合形式，如主导型、AI 引导型以及混合工作坊，保障了教学的多样性与灵活性。表 5 清晰描述了人机协作混合教学模式，有助教学活动的系统规划与执行。

表 5 人机协作混合教学模式
Table 5 Human-AI collaborative hybrid teaching model

教学环节	教师职责	AI 系统职责	协作方式
课前准备	制定目标、规划课程活动	个性化预习资料推送	教师审核 AI 材料
	准备核心内容	知识点分析	调整计划
课堂教学	讲授核心概念	实时代码示例、基础问题答疑	教师与 AI 同步支持
	组织讨论与答疑	行为数据记录	学生可交互提问
课后延伸	总结反思	针对性练习生成、自动批改	AI 反馈辅助教学
	个性化指导	学习报告	教师针对性干预
项目实践	指导项目方向	代码质量检测、调试支持	人机协同完成项目
	创新能力培养	模拟环境反馈	联合评估成果

项目驱动与协作式编程将理论学习与实践应用紧密结合，通过真实项目和团队协作培养学生的工程思维与职业素养。人机协作的混合教学模式则实现了教学过程的个性化与精准化，在课前、课中、课后形成完整学习闭环，AI 系统与教学设计者优势互补，共同提升教学质量与效率。这两种方法相辅相成，既强化了学生的实践能力与协作意识，又提供了个性化的学习体验与精准指导，共同构建了适应智能时代的程序设计教育新范式。随着 AI 技术的不断发展，这些教学模式将持续优化，为培养未来所需的创新型编程人才提供有力支持。

4 人工智能赋能的程序设计课程教学的风险防控与平衡策略

4.1 技术依赖与思维能力弱化风险

AI 编程助手提供的即时代码生成和自动完成功能，在提高学习效率的同时，也可能导致学生对 AI 工具的过度依赖。教师担忧学生可能过度依赖 AI 生成代码而忽视深入理解程序设计原理。这种依赖表现为，学生未经深入思考就直接采纳 AI 建议；遇到问题首先求助 AI 而非独立分析；缺乏对编程问题的耐心和韧性。长期而言，这种依赖可能削弱学生的算法思维、逻辑推理和创新解决问题的能力。

AI 系统快速生成代码的能力可能导致学生对底层计算原理和算法设计的忽视。当学生习惯于获取现成解决方案时，对数据结构选择、算法复杂度分析和内存管理等核心概念的理解可能变得肤浅。这种“黑箱化”学习方式造成的知识断层，将严重影响学生应对复杂编程挑战的能力基础。

此外，AI 系统生成的代码往往遵循特定范式和最佳实践，长期接触这些“标准解法”可能导致学生形成思维定式，缺乏探索多元解决路径的意愿和能力。创新思维的受限将影响学生在面对非常规问题时的表现，不利于培养程序设计的创造力。这种影响在程序设计基础阶段尤为明显，可能导致学生在后续学习中遇到系统性困难。

4.2 平衡自动化与基础训练

建立“AI 使用阶梯”机制，根据学习阶段控制 AI 辅助程度，是确保基础能力培养的有效策略。这一机制的核心是在不同学习阶段合理设置 AI 工具的使用范围和方式，实现自动化便利与基础训练的动态平衡。

具体实施如表 6 所示，初学阶段（大一），适当限制 AI 功能，强化基础知识学习和手动编程训练，要求学生完成基础算法和数据结构的手写实现；进阶阶段（大二、大三），逐步开放代码补全和错误检测等辅助功能，但要求学生解释 AI 生成的代码并进行改进优化；高级阶段（大四），鼓励

学生将 AI 作为协作伙伴，共同应对复杂项目挑战，培养 AI 赋能下的高阶问题解决能力。

表 6 大学不同阶段 AI 工具使用建议

Table 6 Suggestions for using AI tools at different stages of college

学习阶段	AI 功能限制	学习重点	教学策略
大一阶段	仅允许语法参考和错误提示	编程基础知识	手写代码训练为主
	禁用代码生成和自动补全	核心语法结构	基础算法手动实现
	限制在实验课后使用	基本算法设计与实现	逐行代码分析与讲解
大二阶段	开放代码补全功能	数据结构设计	AI 工具使用启蒙教育
	允许有限的错误修正	算法优化思想	AI 辅助编程实践
	限制完整代码生成	模块化设计能力	代码评析与优化
大三阶段	开放大部分 AI 功能	软件工程实践	算法效率分析
	鼓励对 AI 建议的批判思考	复杂问题分解	AI 生成代码批判性学习
	要求改进 AI 生成内容	协作开发能力	AI 协作解决复杂问题
大四阶段	全面开放 AI 功能	创新应用开发	中大型项目开发
	鼓励创造性应用	前沿技术探索	多方案比较与选择
	探索 AI 工具的创新用法	专业素养与职业能力	创新项目孵化
			研究性学习
			职业场景模拟

5 结论

本研究系统探讨了 AI 技术在程序设计课程中的应用模式与价值。AI 赋能程序设计教学不仅能够优化教学流程、丰富教学资源，更能转变传统的教学范式，促进学生编程思维和创新能力的培养。基于“人机协同”的理念构建的教学框架，为程序设计课程的改革提供了可操作的路径。

AI 技术在程序设计教学中的应用需要处理好技术与教学的关系，避免过度依赖 AI 而忽视基础能力培养。教师角色也需要从知识传授者转变为学习引导者和 AI 应用的设计者。这种转变不仅涉及教学理念的更新，也对教师的技术素养提出了更高要求。

随着 AI 技术的不断发展，程序设计教学将迎来更多可能性。在应用过程中，需要不断优化 AI 辅助教学的策略和方法。未来研究可以从以下两个方面展开：

- (1) 建立更精细化的 AI 辅助程序设计教学评估体系，通过纵向追踪研究验证长期学习效果；
- (2) 深入研究不同类型 AI 工具在编程各阶段学习中的差异化应用策略，特别是如何针对不同编程概念难点设计适配性 AI 辅助方案。

总之，AI 与程序设计教学的深度融合，将为培养适应数字时代的创新型技术人才提供有力支撑，推动编程教育向更高质量、更具个性化的方向发展。

参考文献

[1] 牟玉亭, 龙 寰, 蒋 浩. 融入 AI 大模型的计算机程序设计教学实践[J]. 电气电子教学学报, 2024, 46(04): 128-131.

[2] 穆 肃, 陈孝然, 周德青. 生成式人工智能赋能教学设计分析: 需求、方法和发展[J]. 开放教育研究, 2025, 31(01): 61-72.

[3] 裴 榕. 生成式人工智能赋能教育教学: 变革影响、风险挑战与实践路径[J/OL]. 当代教育论坛, 1-10[2025-04-09].

[4] 颜丽蓉, 储节旺, 李振延, 等. 生成式人工智能融入信息资源管理学科专业课程教学的路径探索研究[J]. 图书馆杂志, 2025, 44(01): 128-138+157.

[5] 沈 苑, 汪 琼. 生成式人工智能支持教学决策的实践困境与关键进路[J]. 电化教育研究, 2024, 45(11): 92-99.

- [6] 张玉柳, 罗江华. 生成式人工智能增强学科教学适应性的逻辑理路与实践路径[J]. 电化教育研究, 2024, 45(11): 100-107.
- [7] 袁金凤. 基于“三链融合”的创新实践型人才培养体系探索——以计算机专业为例[J]. 实验室研究与探索, 2025, 44(02): 139-146.
- [8] 李 锋, 袁雨欣, 顾小清. 智能时代编程教育如何培养学生的创新能力——基于编程项目活动联通“学编程”与“用编程创新”[J]. 现代远程教育研究, 2023, 35(06): 11-18.
- [9] 丁美荣, 王同聚. 人工智能教学中“知识建构、STEM、创客”三位一体教学模型的设计与应用[J]. 电化教育研究, 2021, 42(04): 108-114.
- [10] 任 浩, 叶常春, 肖 政. 任务驱动教学法在程序设计课程中的实践与分析[J]. 计算机工程与科学, 2019, 41(S1): 109-113.
- [11] 王明芳. 基于元认知策略的《Java 程序设计》课堂教学研究[J]. 教育理论与实践, 2015, 35(06): 52-53.
- [12] 张彦芳, 高 璐, 李 艳. AI 技术项目驱动下的程序设计类课程教学模式创新[J]. 计算机教育, 2025, (02): 159-163.
- [13] 苏小红, 苗启广, 陈文字. 基于 AI 赋能和产教融合提升程序设计能力的个性教学模式[J]. 中国大学教学, 2023, (06): 4-9.

基金项目: 江苏省教育科学规划课题重点课题《人工智能赋能学习情绪分析及干预研究》
(B-b/2024/01/166)

¹ **第1作者简介:** 徐琴(1987-), 女, 博士, 副教授, 研究方向: 信息化教育教学, 网络拥塞控制。

E-mail: missxuqin@jit.edu.cn

*** 通讯作者简介:** 张迁(1994-), 男, 博士, 讲师, 研究方向: 信息化教育教学, 计算机视觉, 深度学习。 E-mail: zhangqian@jsou.edu.cn