

基于计算机视觉双项目实践的本科生科研能力培养路径探析

卜彤^{1,a,*}

¹ 西安工业大学, 陕西, 西安, 中国

^a 109226489@qq.com

摘要: 本文结合我在本科阶段参与的两个计算机视觉项目, 整理科研入门时遇到的问题和处理过程。两个项目分别涉及创造性图像生成和物联网实时街景分割, 我主要负责模型搭建、训练调试和结果整理。开始时, 我尝试在个人电脑上完成训练, 但显存不足、等待时间长等问题影响了实验进度, 后来将主要训练任务转到云端 GPU 服务器。使用服务器后, 又需要学习环境配置、数据上传和日志保存。模型能够运行以后, 还要对照输入图像检查结果, 再决定如何修改。文中结合保存下来的实验数据, 说明分割模型不同配置的结果, 以及图像生成效果与运行时间之间的取舍。这些经历让我对模型原理有了更具体的认识, 也让我开始重视实验记录和结果说明。本文主要是个人实践总结, 所列技术指标不用于衡量学生科研能力的高低。

关键词: 计算机视觉; 创造性图像生成; 物联网实时街景分割; 云端 GPU 服务器

基金信息: 西安工业大学国家级大学生创新创业训练项目 (202510702030)、西安工业大学校级大学生创新创业训练项目 (X202510702184)

An Analysis of the Pathway for Cultivating Undergraduate Research Skills Based on Computer Vision Dual-Project Practice

Tong Bu^{1,a,*}

¹ Xi'an Technological University, Xi'an, Shaanxi, China

^a 109226489@qq.com

Abstract: This paper describes my experience in two undergraduate computer vision projects: creative image generation and real-time street-scene segmentation for Internet of Things applications. I mainly worked on model implementation, training, debugging, and organizing results. I first tried to train the models on a personal computer. Limited GPU memory and long training times led me to move the main training jobs to rented cloud GPUs. I then had to learn how to set up the environment, upload data, and save logs. Once the models could run, I compared their outputs with the input images to decide what to change. Using saved experiment records, the paper discusses the results of different segmentation configurations and the trade-off between image generation results and running time. These tasks helped me understand the models more clearly and pay more attention to keeping records

and explaining results. This is an account of my own experience; the technical metrics are not measures of students' research skills.

Keywords: Computer Vision; Creative Image Generation; Real-time Street Scene Segmentation of the Internet of Things; Cloud-based GPU Servers

1 引言

刚接触计算机视觉项目时，我已经在课程和资料中学过一些深度学习知识，但真正开始实验后，仍然有不少事情不会做。除了理解网络结构，还要处理数据读取、软件安装和结果保存。深度学习、注意力机制等研究是学习这些模型的基础[1-2]；关于本科生科研训练的文献，也讨论了学生在实际研究任务中的学习过程[19-20]。我想结合自己的经历，说明从学习理论到动手做实验，中间还会遇到哪些问题。

我在本科阶段参与了《智能艺术创造力：基于人工智能技术的创造性图像生成技术研究》和《物联网智能实时街景监测识别系统：基于改进分割模型的高效街景图像分割技术研究》。前一个项目需要生成或转换图像，后一个项目需要识别街景图像中的道路、车辆和行人等目标。两项任务不同，但都离不开模型搭建、数据准备和训练调试。我在两个项目中遇到过一些相似的问题，也逐渐发现，判断结果好坏时不能用同一种标准。

我的工作主要是搭建模型、运行训练和整理实验结果。最初，我想用自己的电脑完成这些工作。实际训练时才发现，显存和计算速度都有限，修改参数以后，往往还要等较长时间才能看到结果。实验做得慢，也就很难及时判断刚才的修改有没有用。

后来，我开始学习使用第三方云端 GPU 服务器。代码和数据传上去后，还需要重新配置虚拟环境、PyTorch 和相关依赖[18]，再检查训练脚本和输出路径。这些操作起初并不熟悉，不能把文件传好就直接开始训练。

查看结果时，两个项目的区别比较明显。生成图像要看主体有没有变形、细节是否自然；分割结果要看类别有没有分错，车辆和行人的边界是否清楚。程序运行结束以后，这些检查仍然要继续做。

下面先介绍两个项目和我的具体工作，再回顾环境搭建、训练迁移和模型修改的过程，最后谈一谈自己的收获。文章主要根据个人经历和保存的实验记录整理，没有开展学生群体的教学对比。

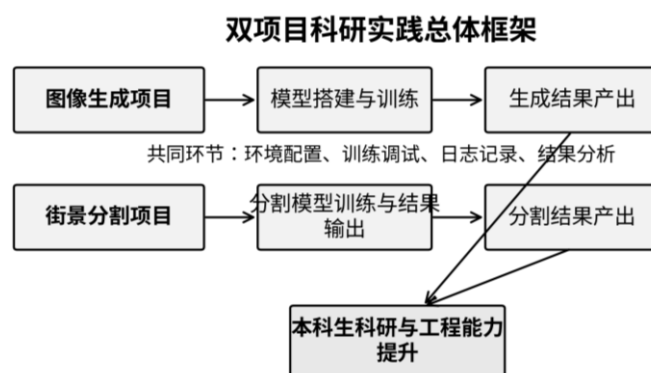


图1. 双项目科研实践总体框架图

2 双项目实践基础与个人任务定位

2.1 创造性图像生成项目概况

创造性图像生成项目的任务是处理输入图像，得到符合要求的新图像。输出既要有预期的变化，也要保留主体轮廓和位置关系。我阅读了生成模型、神经风格迁移、感知损失，以及 Transformer 和扩散模型等方面的资料[3-9]，帮助自己理解这一方向。文献中涉及的方法很多，项目并没有把它们全部实现一遍。

我先学习代码如何读取图像、提取特征和输出结果，再安装 Python、PyTorch 及项目所需的依赖。模型运行起来以后，我开始把注意力放到生成图像上，查看主体有没有变形、局部有没有伪影。有时程序没有报错，但输出仍然不符合要求，还需要继续检查。

有些输出变化很明显，却破坏了原来的主体；有些保留了内容，但看不出足够的变化。遇到这种情况，我会对照输入和输出，再检查参数及处理过程。单看一张图，很难判断问题到底出在哪一处。

2.2 物联网实时街景分割项目概况

街景分割项目需要给图像中的像素分配类别，例如道路、车辆、行人和建筑。为了理解模型如何完成这项任务，我阅读了 FCN、U-Net、PSPNet 和 DeepLab 等相关研究[10-15]，并参考 Cityscapes 的数据介绍[16]。项目希望用于实时街景监测，但是否能达到实时要求，还要看运行设备和整个系统的处理速度。

我的主要工作是搭建训练流程，弄清原图、标注图和预测掩码之间的关系，并整理模型输出。检查结果时，我会关注道路区域是否连续，车辆和行人的轮廓是否清楚，以及远处目标和遮挡位置有没有漏分。预测图虽然看起来很直观，仍然需要与原图、标注图对照。

在我的本地电脑上，分割训练比较容易受到资源限制。图像尺寸和批量大小增加以后，显存压力更大，训练也更慢。这是我后来学习云端训练的一个直接原因。这里说的是当时使用的设备和设置，不代表所有分割模型都比生成模型更耗资源。

2.3 个人主要工作与实践任务

表 1 列出了我在两个项目中的主要工作。搭建模型时，要先弄清输入输出并配置环境；运行训练时，要设置参数、处理报错和查看结果。训练结束后，还要保存图像、权重和日志，并记下对应的设置。如果这些材料对不上，过一段时间再看，就很难知道当时是怎样得到结果的。

表 1. 双项目实践内容与个人任务定位

项目方向	创造性图像生成方向	物联网实时街景分割方向
任务目标	生成符合任务需求的新图像结果	对街景要素进行像素级识别与分割
核心环节	输入处理、模型搭建、训练调试、生成输出	数据读取、分割模型训练、预测与可视化
本人主要工作	搭建运行环境，训练模型，观察生成质量并整理结果	配置训练流程，完成分割训练，输出掩码和对比结果
结果形式	生成图像、训练日志、模型权重	分割掩码、预测图、可视化结果

最开始，我主要跟着已有资料和代码操作，对模型细节理解得还不够。实验次数增加以后，我才开始根据结果中的问题，回头查看相关代码和参数。两个项目的方案都是边做边调整的，一开始并没有完全想清楚。

3 模型搭建与训练实践过程

3.1 本地环境搭建与初步训练尝试

我先在个人电脑上建立 Python 虚拟环境，安装 PyTorch 和其他依赖，再修改数据路径和参数文件。配置时需要注意软件版本是否匹配，以及程序能否调用 GPU[18]。这一阶段首先要解决的是让代码读到数据，并正式开始训练。

那时我比较关注程序有没有报错，对完整训练需要多少资源考虑得不够。短时间运行成功以后，继续训练仍可能出现显存不足或速度变慢的问题。前面的测试通过了，后面的工作也不能省掉。

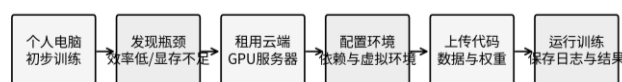
等待结果是本地训练中很直接的困难。输入图像较大、训练轮次较多时，修改一次就要等很久。这样一来，能够尝试的方案也受到限制。我后来才把运行时间一起考虑进去，而不只是看某个方案在原理上是否可行。

显存不足时，我需要调整输入尺寸、批量大小或模型设置。降低这些设置可以先让程序运行，但得到的结果也会受到影响。所以，初步调试用了什么设置，需要另外记下来，后面比较时不能混在一起。

我还遇到过结果文件不好找的问题。不同实验的输出放在一起，名字也没有写清楚，再想比较时，就难以找到对应的参数。这个问题后来换到服务器上同样需要处理。

经过这些尝试，我把本地电脑主要用于检查代码和数据流程，再把较长的训练任务安排到云端。前期虽然花了不少时间配置和调试，但至少让我熟悉了实验从哪里开始、哪些地方容易出错。

从本地训练到云服务器训练的迁移流程



迁移重点：提高实验效率，同时强化环境配置、命令行操作和实验管理能力

图2. 从本地训练到云端 GPU 服务器训练的迁移流程图

3.2 训练效率瓶颈与云端 GPU 服务器环境迁移

开始使用云端 GPU 服务器后,我需要重新检查软件版本、数据位置和输出路径。服务器的资源更充足,但环境与本地电脑不同,原来的代码还需要做相应配置。

远程操作是我首先要学的内容。登录服务器、建立目录、上传文件和运行脚本,都要用到命令行。刚开始,我对运行目录和数据路径不太熟悉,后来逐步把代码、数据、配置和输出分开存放,查找起来才方便一些。

安装依赖时,也会遇到版本不兼容、库无法导入或 GPU 不能调用的问题。我为两个项目分别建立虚拟环境,按代码要求安装 Python、PyTorch、CUDA 相关依赖和其他库。把版本信息记下来以后,重新配置时就不用完全从头摸索。

训练开始以后,我会查看损失变化,检查日志和结果文件有没有正常保存。我也开始记录模型版本、参数和数据路径,尽量让它们与权重文件对应起来。只有输出图而没有过程记录,后面分析时还是会缺少信息。

云端训练减少了部分等待时间,但也多了远程操作和文件管理的工作。数据上传、脚本运行、日志查看以及结果下载,都需要自己检查。用过一段时间以后,我对一项实验从开始到结束要做哪些事,才有了比较清楚的认识。

表 2 整理了本地初调、主要训练和结果整理阶段的工作。由于没有在相同条件下专门测量本地与云端的训练时间,这里只说明各阶段做了什么,不列训练加速倍数。

表 2. 本地训练与云端 GPU 服务器训练的阶段分工

阶段	初步连通性测试	主要训练阶段	后期结果产出
运行环境	个人电脑本地环境	云端 GPU 服务器	云端 GPU 服务器
主要任务	检查输入输出、依赖与路径	保存参数、权重与日志	分割验证 500 张;生成测试 400 组
记录内容	代码运行与报错	训练设置与过程	评价协议、指标与结果图
适用情况	小规模检查与排查	执行主要训练任务	同条件比较、整理实验材料

注:本表描述任务分工;后期评价规模对应 2026 年 8 月归档记录,不表示每个训练阶段均使用相同数据或相同运行条件。

3.3 模型选择、构建与多轮修改

环境基本稳定后,我开始比较模型的输出。生成项目要兼顾图像变化和-content保留,分割项目要关注类别、边界及运行时间。已有研究提供了不少选择[3-17],但用到自己的任务中,仍然需要实际运行和检查。

在生成项目中,我会把输入和输出放在一起查看,注意内容是否保留、变化是否符合要求[3-8]。变化很大时,主体可能已经变形;变化太小时,又可能没有完成任务。除了看训练损失,还需要检查这些具体表现。

分割结果中的问题通常可以在图像上找到位置,例如远处行人没有分出来,或者道路区域出现断裂。我会先对照原图、标注和预测图确认问题,再结合模型的处理方式继续分析[13-16]。图片能显示哪里分错了,但不一定能直接说明原因。

修改模型时,还要考虑电脑或服务器能否承受。结构变复杂以后,即使结果有所改善,也可能需要更多显存和训练时间。生成项目中的一些调整只发生在推理时,不需要重新训练,后面整理结果时也要把这两种情况分清楚。

反复尝试以后,我发现每次修改最好先想清楚要解决哪一个问题。改完再看结果,比较是否符合预期。如果没有改善,就根据这次记录继续排查。这样比只想着让模型“再好一点”更容易安排下一步。

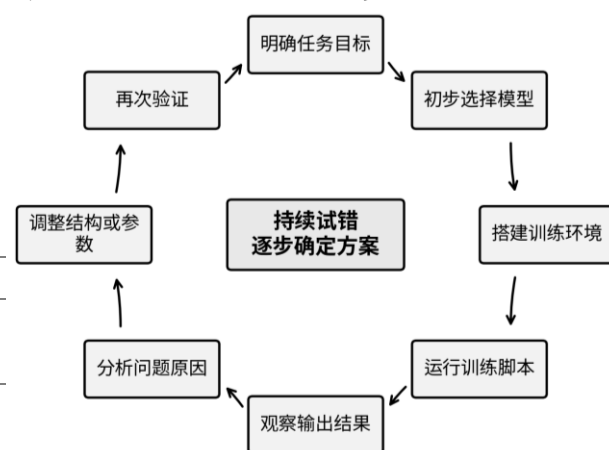


图3. 模型选择与修改的试错闭环图

3.4 训练结果产出与实验材料整理

训练结束以后,还要花时间整理输出。生成项目需要保存输入图像和生成图像,分割项目还要对应保存标注图和预测掩码。整

理时要能找到每张图来自哪次实验，效果不理想的结果也值得留下来分析。

查看生成图像时，我主要关心主体轮廓是否保留，局部有没有变形或伪影，以及输出是否符合任务要求。这些观察比较直观，但带有主观判断。后面加入数值比较，是为了从另一个角度检查结果。

分割结果则需要多看一些场景。普通道路图像上表现较好，不代表在光照复杂、目

标被遮挡或目标很小时也一样。我按场景整理结果，方便回头查看模型容易在哪些地方出错。

整理材料时，我把输出、日志、权重和参数说明尽量放在对应的位置。写报告时需要解释某个结果，就可以回去查记录。下面两表使用的是 2026 年 8 月 11 日和 14 日归档的分割与生成实验数据。

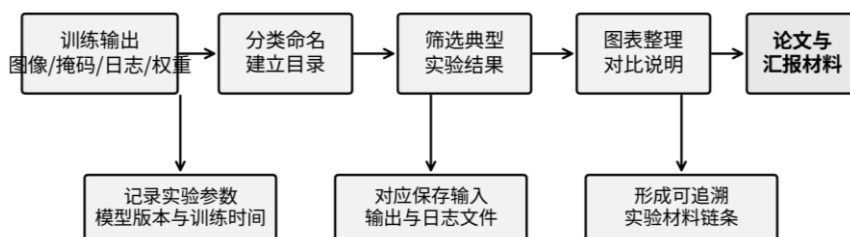


图4. 训练结果产出与成果表达流程图

街景分割记录采用 DeepLabV3+，在 Cityscapes 验证集 500 张图像上按 1024×2048 分辨率评价，边界宽度与匹配容差均为 3 像素。表 3 中的 CE 和 ABL 分别表示交叉熵训练与加入主动边界损失的训练配置，两者各使用三个种子检查点的等权平均模型。SS 表示单尺度推理；HFlip 将原图与水平翻转预测的类别分数平均；SegFix 采用官方偏移文件作边界后处理。三列是实验配置，不是连续训练阶段。ABL 单尺度相对 CE 的 mIoU 提高 0.15 个百分点，组合推理相对 ABL 单尺度提高 3.29 个百分点，后者不能全部归因于 ABL。该表不提供端到端实时性结论。

表 3. 物联网街景分割项目实验指标（归档记录）

实验配置	CE+SS	ABL+SS	ABL+HFlip +SegFix
mIoU/%	68.95	69.10	72.39
PA/%	94.17	94.23	95.20
Boundary F/%	53.95	54.50	62.36
结果说明	训练基线	加入边界 损失	增加翻转与边 界后处理

注：mIoU 为平均交并比，PA 为像素准确率，Boundary F 为边界 F 分数。数据来自 2026 年 8 月 11 日归档结果，均为验证集指标。

表 4 的前 3 行取冻结测试集 20 幅内容图像×20 幅风格图像共 400 组均值，比较 256 与 512 尺度下的输出或内部响应；低频锚定无需重新训练。注意力响应漂移为六层对齐响应距离的平均值，VGG 漂移为输出对齐后

的特征距离，二者越低表示跨尺度差异越小；边缘相关性越高越好。耗时另在一个固定图像组合上预热 10 次、同步计时 50 次后取中位数，仅含模型前向：基线一次高分辨率前向，锚定增加低分辨率分支，不含读写文件。相对变化按两组均值或中位耗时计算。

表 4. 创造性图像生成项目实验记录（归档记录）

指标	原始模型	低频锚定	相对变化
注意力响应漂移	0.1485	0.1040	-29.98%
VGG 漂移	0.2106	0.1948	-7.51%
边缘相关性	0.5733	0.5650	-1.44%
前向耗时/ms	53.46	86.14	+61.15%

注：数据来自 2026 年 8 月 14 日 StyTR-2 跨尺度实验。

4 实践过程中遇到的问题及解决路径

4.1 本地训练效率低下与算力资源不足问题

回头看，我在开始时低估了多轮训练需要的时间。本地电脑做简单检查比较方便，但完整训练往往要等较久。后来我逐步把小规模调试与主要训练分开安排，没有再把所有任务都放在个人电脑上完成。

为了先让程序运行，我尝试过减小输入尺寸、批量大小和训练轮次。这些做法缓解了资源压力，不过也改变了实验条件。因此，哪些结果只用于初调，需要记清楚，以免后来拿来与完整训练的结果直接比较。

主要训练转到云端以后，本地电脑仍然用于部分检查。换用服务器解决了部分算力问题，文件分类和实验记录还要自己做好。

4.2 云端 GPU 服务器环境配置与训练迁移问题

刚接触服务器时,我需要学习实例选择、远程连接、上传和下载文件等操作。一个步骤出错,就可能影响后面的训练。命令行用得多了,我才比较清楚如何查看运行位置和任务状态,遇到问题时也有了排查的方向。

环境报错时,我会先看是哪个库没有导入,还是软件版本或 GPU 调用出了问题。为项目单独建立虚拟环境可以减少干扰,但装好以后仍然要检查。保存版本信息,对以后重新搭建环境也有帮助。

路径出错时,需要回到脚本里核对它实际读取的位置。将代码、数据和输出分开管理后,查找这类问题方便了一些,训练结束后也更容易找到结果。

云端 GPU 按使用时间收费,我因此更注意程序是否在正常训练。除了看它有没有停止,还会检查损失变化和文件保存情况。如果日志已经显示异常,就需要先排查,不能一直等待程序自己结束。

4.3 模型选择与结构修改中的反复试错问题

选择模型时,我曾希望找到一个现成方案就能完成任务。实际做下来发现,仍然要看当前数据和输出效果。后来我会先指出图像中哪里不对,再考虑改结构、参数,还是输入处理方式。模型更复杂,并不保证结果更合适。

生成实验中,有些变化是有代价的。表 4 里的低频锚定使注意力响应漂移下降 29.98%、VGG 漂移下降 7.51%,但边缘相关性下降了 1.44%,中位前向耗时也从 53.46 ms 增加到 86.14 ms。同期记录中,内容损失增

加约 0.45%,风格损失降低约 0.45%。所以,这个方案让不同分辨率下的结果更稳定,却不能简单说成画质全面变好了,还要看任务能否接受细节变化和额外耗时。

分割数据也要分开理解。表 3 中,CE 单尺度的 mIoU 是 68.95%,加入 ABL 后是 69.10%;再使用 HFlip 和 SegFix,才达到 72.39%。前一次比较涉及训练设置,后一次增加了推理处理。整理结果时,需要把这些区别交代清楚,不能把所有提高都说成是某一个改动带来的。

实验次数多了以后,记录尤其重要。如果一次改了很多地方,又没有记清楚,看到结果变化也很难解释。这也是我后来更注意说明每次改了什么、为什么改的原因。

4.4 训练结果输出与实验材料整理问题

随着实验增加,图像、权重和日志越来越多。早期文件名不清楚的问题,在回头整理时就比较麻烦。我后来按实验批次和模型设置分类,尽量把结果与当时的条件一起保存。

生成图像需要和输入、模型设置对应。否则,到了展示结果时,虽然能看到图像,却说不清是哪次实验得到的。分类保存以后,查找和比较都方便一些。

分割项目中的原图、标注图和预测图也要一一对应。我将它们按场景整理,再放在一起查看。这样比较容易发现哪些目标漏分了,或者哪些类别混在了一起。

整理这些材料以后,我发现写结果说明也依赖前面的记录。输入、参数和输出保存得完整,解释起来才有依据。遇到材料不足的地方,需要回去查清楚,不能只根据一张效果图作判断。

表 5. 实践过程中主要问题及解决路径

问题类型	具体表现	解决路径	能力提升
本地算力不足	训练速度慢、显存受限、长时间运行不稳定	降低批量大小进行初调,后续迁移到云端 GPU 服务器训练	资源评估与训练效率意识
环境配置复杂	依赖版本不兼容、GPU 调用异常、路径错误	使用虚拟环境隔离项目,规范安装依赖和管理目录	环境配置与问题排查能力
模型方案难确定	不同结构效果差异大,训练耗时和效果之间需要平衡	围绕任务目标进行多轮训练、观察、对比和修改	科研试错与模型分析能力
结果材料混乱	图像、日志、权重文件数量多,难以追溯实验过程	按实验批次分类命名,保存参数、日志与输出结果	实验管理与科研表达能力

5 科研能力培养效果分析

5.1 从课程学习到模型实践的能力转变

参与项目以前,我对深度学习的认识主要来自课程、公式和示例。真正操作时,要自己处理数据输入、参数设置和程序报错。一些原先觉得已经学过的内容,到了代码里才发现还没有理解清楚,需要重新查资料。

两个项目让我从不同方面理解模型。生成图像要看原有内容还在不在,变化是否合适;分割图像要看类别和位置是否正确。同样是输出一张图,检查的重点差别很大。实际对照结果,比只看模型结构图更容易理解任务的要求。

我也开始关心数据怎样进入模型,损失函数怎样参与训练,以及结果如何保存。这些内容原先在学习时比较分散,做实验后才逐渐联系起来。这里说的是我对自己学习过程的回顾,没有另外做能力测评。

5.2 从本地调试到云端训练的工程能力提升

在工程操作方面,我学到的不只是如何连接服务器。配置依赖、检查路径、查看日志和下载结果,都需要自己完成。以前只关注训练命令能否执行,后来才知道,前后的准备和检查也会影响实验能否顺利完成。

从自己的经历看,这些基础操作值得在科研入门时多练习。例如,程序为什么没有用到 GPU,数据读到了哪里,权重保存在什么位置,都需要弄清楚。遇到报错以后,能先检查这些地方,至少可以缩小问题的范围。

本科生科研训练相关文献也讨论了在实际任务中学习的过程[19-20]。我在项目中的体会是,有些操作只看说明很难记住,真正遇到问题并处理以后,理解会具体一些。这只是个人体会,还不能说明哪一种教学安排更有效。

5.3 从单次运行到多轮试错的科研思维提升

最开始,我希望代码能正常运行并得到输出。后来做了几轮修改,才发现还要说明结果为什么变化,以及这次修改有没有解决

原来的问题。即使实验失败,留下报错信息和参数记录,也能为下一次排查提供帮助。

生成项目中,常会遇到两个结果各有优缺点的情况。一个保留内容较好,另一个变化更明显,很难只凭第一眼印象作决定。最后还是要回到任务要求,看哪一种结果更合适。

分割实验让我注意到,指标也有适用的范围。表3是在500张验证图像上得到的结果,还不能说明模型换到真实部署环境后会怎样。报告数值时,需要一起写明测试条件。

对于刚开始做实验的同学,我觉得可以先练习写一小段实验说明,把这次改了什么、结果如何、还有哪里不明白记下来。篇幅不必很长,但以后回看时应该能读懂。这样也方便与老师讨论下一步怎么做。

5.4 从结果产出到成果表达的科研写作能力提升

整理项目材料时,我才更明显地感到,得到输出和写清楚结果是两件事。图像需要说明输入和设置,指标需要交代测试条件。有时写到这里,才发现前面的记录不完整,只能再回去核对。

已有研究讨论了本科生科研经历及其学习价值[19-20]。对我来说,写作中的一个收获是学会检查自己的说法有没有依据。如果找不到相应材料,就需要查清楚,或者把结论写得更谨慎。

我以前比较容易按先后顺序写“运行了模型”“得到了结果”“又进行了修改”。现在会多解释为什么要改,以及结果说明了什么。自己目前在这方面还需要练习,但已经比单纯记录操作多考虑了一步。

6 结语

通过两个项目,我接触了图像生成和街景分割,也经历了从配置环境到整理结果的过程。对我影响比较大的,是发现科研中有不少看起来基础、却不能省略的工作。路径没有设好,训练就无法开始;实验记录不清楚,后面也很难比较结果。

对刚开始参加项目的本科生,我的建议是先完成小规模运行,熟悉代码和数据,再

安排较长的训练。实验过程中及时保存参数、日志和输出，写报告时会少一些来回查找的困难。遇到不理想的结果，也可以先把问题描述清楚，再讨论如何修改。

本文只总结了我参与两个项目的经历，没有对照组，也没有用统一量表测量能力变化。表 3 和表 4 说明的是模型实验结果，不能用来证明学生的科研能力提高了多少。分割结果和生成耗时也都有各自的测试条件。以后如果要进一步研究这种培养方式，还需要记录更多学生的学习过程，并采用明确的评价标准。

参考文献

- [1] LeCun Y, Bengio Y, Hinton G. Deep learning[J]. Nature, 2015, 521(7553): 436-444.
- [2] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[C]//Advances in Neural Information Processing Systems. 2017: 5998-6008.
- [3] Goodfellow I, Pouget-Abadie J, Mirza M, et al. Generative adversarial nets[C]//Advances in Neural Information Processing Systems. 2014: 2672-2680.
- [4] Gatys L A, Ecker A S, Bethge M. Image style transfer using convolutional neural networks[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016: 2414-2423.
- [5] Johnson J, Alahi A, Fei-Fei L. Perceptual losses for real-time style transfer and super-resolution[C]//European Conference on Computer Vision. Cham: Springer, 2016: 694-711.
- [6] Huang X, Belongie S. Arbitrary style transfer in real-time with adaptive instance normalization[C]//Proceedings of the IEEE International Conference on Computer Vision. 2017: 1501-1510.
- [7] Deng Y, Tang F, Dong W, et al. StyTr2: Image style transfer with transformers[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022: 11326-11336.
- [8] Rombach R, Blattmann A, Lorenz D, et al. High-resolution image synthesis with latent diffusion models[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022: 10684-10695.
- [9] 陈淑环, 韦玉科, 徐乐, 等. 基于深度学习的图像风格迁移研究综述[J]. 计算机应用研究, 2019, 36(08): 2250-2255.
- [10] Long J, Shelhamer E, Darrell T. Fully convolutional networks for semantic segmentation[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015: 3431-3440.
- [11] Ronneberger O, Fischer P, Brox T. U-Net: Convolutional networks for biomedical image segmentation[C]//Medical Image Computing and Computer-Assisted Intervention. Cham: Springer, 2015: 234-241.
- [12] Zhao H, Shi J, Qi X, et al. Pyramid scene parsing network[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017: 2881-2890.
- [13] Chen L C, Papandreou G, Kokkinos I, et al. DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2018, 40(4): 834-848.
- [14] Chen L C, Papandreou G, Schroff F, et al. Rethinking atrous convolution for semantic image segmentation[EB/OL]. arXiv:1706.05587, 2017.
- [15] Chen L C, Zhu Y, Papandreou G, et al. Encoder-decoder with atrous separable convolution for semantic image segmentation[C]//European Conference on Computer Vision. 2018: 801-818.
- [16] Cordts M, Omran M, Ramos S, et al. The Cityscapes dataset for semantic urban scene understanding[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016: 3213-3223.
- [17] Chollet F. Xception: Deep learning with depthwise separable convolutions[C] // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017: 1251-1258.
- [18] Paszke A, Gross S, Massa F, et al. PyTorch: An

imperative style, high-performance deep learning library[C]//Advances in Neural Information Processing Systems. 2019: 8024-8035.

- [19] Linn M C, Palmer E, Baranger A, et al. Undergraduate research experiences: Impacts and opportunities[J]. Science, 2015, 347(6222):

1261757.

- [20] National Academies of Sciences, Engineering, and Medicine. Undergraduate research experiences for STEM students: Successes, challenges, and opportunities[M]. Washington, DC: The National Academies Press, 2017.