### 以代码为笔，融视觉为诗——我的多模态大模型探索之旅

**王景轩**

**智实2201班**

**U202215302**



在本次“以图生诗”的视觉与自然语言处理课程项目中，我承担了项目整体架构设计与绝大多数的代码编写工作。我的主要工作贯穿了项目的全生命周期，具体包括：独立设计并实现了一套基于Gemini-2.5-Pro API的自动化数据标注流水线，通过精密的提示词工程为项目高质量完成了全部2369张图片的标注任务；主导了技术选型，决定采用参数高效的LoRA方法，并依托LLaMA-Factory框架成功对Qwen2.5-VL-7B模型进行了监督微调，显著提升了模型在诗歌生成领域的专业能力；与卢川柱同学协作完成了最终“三模型对比”Web UI的搭建，我重点负责后端逻辑集成与多模型调度；此外，我还积极参与了最终项目报告的技术部分修订与初版演示视频的录制工作。可以说，我推动了项目从0到1的技术突破，并保障了从数据、模型到最终产品呈现的完整技术链路的实现。

随着人工智能技术的浪潮席卷而来，视觉与自然语言处理的交叉领域——多模态学习，正以其独特的魅力吸引着无数探索者。当本学期的《视觉与自然语言处理》课程发布期末作业，要求我们利用视觉语言大模型构建一个兼具实用性与创造性的应用时，我内心充满了激动与期待。这个项目不仅是一次课程考核，更是一个将理论知识付诸实践、亲手触碰前沿技术的绝佳机会。我们小组最终确立了“以图生诗”这一富有挑战性与人文气息的课题，旨在探索如何让冰冷的代码理解图像中的意境，并生成蕴含中国古典韵味的诗词。在这个过程中，我非常有幸地承担了项目绝大多数的技术实现工作，从底层数据流水线的设计，到核心模型的微调，再到最终前端界面的整合，这段经历让我对多模态大模型有了前所未有的深刻理解。

**项目的起点：构建高质量数据集的自动化流水线**

项目伊始，我们面临着一个核心且棘手的难题：数据。课程组提供了包含2369张图片的丰富数据集，但这些图片没有任何标注。对于“以图生诗”这样的生成任务而言，高质量、大规模的“图像-诗歌”配对数据是模型成功的基石。手动标注不仅耗时巨大，而且质量难以保证，不同的人对同一张图片可能会有截然不同的诗意解读。因此，我从一开始就确立了核心思路：我们必须利用现有的、最强大的商业闭源大模型，通过精巧的工程设计，构建一条自动化的、高质量的数据标注流水线，为后续我们自己模型的训练奠定坚实的基础。

我将目光投向了当时多模态能力最为出色的模型之一——Google的Gemini-2.5-Pro。我判断，其强大的图像理解和文本生成能力，足以胜任“看图作诗”这一复杂的创造性任务。然而，简单地向模型抛出一张图片，让其自由发挥，是远远不够的。为了保证生成数据的结构化、一致性和高质量，我投入了大量精力进行“提示词工程”（Prompt Engineering）。我深知，一个好的Prompt，是与大模型高效沟通的桥梁。

为此，我精心设计了两个核心文件：prompt.txt 和 instructions.txt。在 instructions.txt 中，我系统性地整理了中国古典诗词的格律要求，包括五言/七言绝句、五言/七言律诗以及各种常见词牌的平仄、对仗、韵脚等规则。这相当于为模型提供了一本详尽的“诗词创作教科书”。而在 prompt.txt 中，我则扮演了一位“AI训练师”的角色。我明确定义了模型的身份——“一个专注于根据图片创作中国古典诗词的AI专家”，并设计了一个极为严格的JSON输出格式。这个JSON结构包含了几个关键字段：create\_condition 用于判断图片是否适合创作诗歌，poem\_type、poem\_title、poem\_content 分别对应诗歌的体裁、标题和内容，而 poem\_explanation 则要求模型对生成的诗歌进行一番解析，这不仅能验证模型是否真正理解了诗歌的意境，也极大地丰富了标注数据的维度。这种结构化的输出设计，是我后续能够顺利进行数据清洗和格式转换的关键所在。

有了精巧的“设计图纸”，下一步就是用代码将其实现为一条自动化运转的“生产线”。我独立编写了核心脚本 generate.py。这个脚本并非简单的循环调用API，我为其构建了强大的工程鲁棒性。它能够自动遍历课程提供的Train和Val两个目录下的所有图片，为了提高效率和稳定性，脚本会预先将公用的prompt.txt和instructions.txt上传。在处理每一张图片时，脚本会调用Gemini-2.5-Pro API，并将返回的、包裹在代码块中的JSON字符串进行精确提取和解析。考虑到API调用可能出现的网络波动或服务不稳定，我加入了异常处理机制，确保单张图片的失败不会中断整个流程。更重要的是，我设计了检查点（Checkpoint）机制，每处理10张图片就自动保存一次阶段性成果。这个看似微小的设计，在面对长达数小时乃至数天的标注任务时，被证明是至关重要的，它避免了因意外中断而导致前功尽弃的风险。

经过数日的连续运行，这条由我一手搭建的自动化数据标注流水线成功处理了全部2369张图片。最终，我们获得了2345条高质量的结构化标注数据，并成功识别并分离了24张不适合进行诗歌创作的图片（例如内容过于简单或现代的截图）。为了验证和分析这批来之不易的数据，我还编写了辅助脚本 count.py，用于统计数据总量、检查是否存在重复的图片路径以及生成失败的记录，确保了数据集的纯净性。可以说，这一阶段的工作，为我们整个项目的成功打下了最为坚实的地基。

**核心突破：开源模型的监督微调与性能飞跃**

拥有了高质量的“教科书”，下一步就是培养我们自己的“学生”。项目的核心目标之一，是探索如何利用我们标注的数据，将一个强大的开源多模态大模型，训练成一个专精于“以图生诗”领域的专家模型。这不仅是课程的要求，也是我个人最感兴趣的挑战。

在技术选型上，我选择了当时备受瞩目的Qwen2.5-VL-7B作为我们的基础模型。它出色的中英文双语能力和视觉语言交互性能，使其成为一个理想的起点。而在微调方法上，我没有采用传统的全量微调（Full Fine-Tuning），因为它对计算资源的要求极为苛刻。经过深入调研，我决定采用监督微调（Supervised Fine-Tuning）中的LoRA（Low-Rank Adaptation）方法。LoRA作为一种参数高效微调（Parameter-Efficient Fine-Tuning, PEFT）技术，其核心思想是在原有大模型的参数矩阵旁边，增加一个低秩的“旁路矩阵”进行训练。这样做的好处是，我们无需改动和重新训练庞大的原始模型参数，只需训练这个新增的、参数量极小的旁路矩阵即可。这使得我们能够在有限的资源下，高效地让模型学习到特定任务的知识，是一种兼具效果与效率的理想方案。

为了实施这一方案，我选择了业界知名的开源项目LLaMA-Factory。它集成了一百多种大模型的统一高效微调流程，极大地简化了训练过程的复杂性。然而，工具的便利不代表工作的轻松。我首先需要将我们由Gemini生成的数据，转换为LLaMA-Factory所要求的特定格式。为此，我编写了 transform.py 脚本。这个脚本负责读取我们之前生成的JSON文件，将其中的“图片-诗歌-解析”三元组，巧妙地转换为模型训练所需的“user-assistant”对话格式。在“user”部分，我将图片信息嵌入，并提出作诗要求；在“assistant”部分，则将诗歌内容和解析作为模型的学习目标。同时，脚本还负责处理图片路径，将其转换为模型可以读取的绝对路径，并支持将新转换的数据追加到已有的训练集中，方便我们进行增量实验。

在准备好数据和训练环境后，我便开始了紧张而兴奋的模型微调工作。我仔细配置了训练参数，包括学习率、训练轮次（Epochs）、LoRA的秩（Rank）等，并在服务器上启动了训练任务。在漫长的训练过程中，我密切监控着模型的训练损失（Training Loss）和评估损失（Evaluation Loss）曲线。当看到两条曲线都稳步下降并最终收敛到一个较低的水平时，我知道，我们的模型正在从数据中汲取养分，逐渐从一个“通才”蜕变为一个“专才”。

训练完成后，一个专属于我们项目的、经过LoRA微调的Qwen2.5-VL-7B模型——Qwen2.5-VL-7B\_LoRA\_image2poem——诞生了。为了让这份成果能够被更多人看到和使用，我将其开源至Hugging Face社区。这不仅是对开源精神的致敬，也是对我个人工作成果的一份珍贵记录。

**终章：系统集成、前端呈现与价值验证**

模型训练的成功只是完成了“从0到1”的构建，如何将其“从1到N”，让用户能够直观地体验和比较，是项目的最后一道关卡。一个没有用户界面的算法，其价值是难以被感知的。因此，我与卢川柱同学合作，共同完成了前端界面的搭建与修改工作。

在这部分工作中，我主要负责后端的逻辑实现和系统集成，也就是 UI\_Web.py 脚本的编写。这是一个基于Gradio框架的Web应用，其核心功能是我设计的“三模型对比”界面。我希望用户能够上传一张图片，然后一键看到三个不同模型的表现：我们赖以构建数据集的强大闭源模型Gemini-2.5-Pro，作为基线的Qwen2.5-VL-7B原始模型，以及我们倾注心血微调得到的Qwen2.5-VL-7B LoRA模型。

实现这个功能需要克服不少技术细节。我需要在一个应用中，同时集成对Google API的调用逻辑，以及本地加载两个Hugging Face模型（原始模型和附加了LoRA权重的微调模型）的逻辑。我编写了统一的predict函数，通过传入不同的模型标识，调度不同的推理后端。对于本地模型，代码需要处理模型的加载、处理器的初始化、以及将用户上传的图片转换为模型可以接受的输入格式。整个过程我力求代码的整洁与高效，确保用户能够获得流畅的体验。

最终的成品效果令人振奋。通过这个界面，我们可以非常直观地看到：原始的Qwen2.5-VL-7B模型虽然也能生成一些与图片相关的文字，但往往流于表面描述，缺乏诗歌的韵味和意境；而经过我们微调后的模型，其生成的诗歌在格律、意象和情感表达上都有了质的飞跃，其效果甚至在很多场景下能够媲美Gemini-2.5-Pro。这种强烈的对比，雄辩地证明了我所构建的数据流水线和采用的微调策略的巨大成功。

除了上述三项核心的技术工作，我还积极参与了项目的其他环节。在项目后期，我录制了初版的演示视频（Demo），向老师和同学们展示了我们系统的完整功能和出色效果。同时，我也深度参与了最终项目报告的修改工作，特别是对技术细节部分的描述，确保了报告的准确性和专业性。

**反思与展望**

回首整个项目历程，我深感收获巨大。这不仅仅是关于技术的磨练，更是关于如何将一个模糊的想法，通过系统性的工程方法，一步步变为现实的宝贵经验。我从中学到的，远不止是某个API的调用方法或某个框架的使用技巧。

我学会了如何进行严谨的提示词工程，理解了与大模型“对话”的艺术；我亲手实践了参数高效微调的强大威力，见证了数据和算法如何共同赋予模型新的生命；我体验了从后端模型到前端应用的全栈开发流程，懂得了技术最终要以用户可感知的方式呈现其价值。从generate.py的自动化标注，到transform.py的数据转换，再到UI\_Web.py的系统集成，每一行代码都承载着我的思考与付出。

当然，项目中也存在可以进一步完善的地方。例如，我们的数据集规模虽然达到了两千多张，但如果能引入更多样化的图片类型（如抽象画、水墨画），或许能让模型的泛化能力和创造力更上一层楼。此外，在诗歌生成的评估上，目前我们主要依赖人工主观判断，未来可以探索引入一些量化的评估指标，如与人类创作的相似度、格律的准确率等。

总而言之，这次期末项目是我在人工智能领域学习道路上的一座重要里程碑。我为自己能够作为核心技术贡献者，推动项目从零到一的实现而感到自豪。这段经历不仅极大地提升了我的代码实现能力和工程实践能力，更让我坚定了在多模态人工智能领域继续探索下去的决心。未来，我希望能继续深入研究，探索如何让AI更好地理解和创造美，让技术与人文碰撞出更加绚烂的火花。