

# 北京郵電大學

## 本科 毕业 设计（论文）



题目： 基于 LLM 的交互式多模态图像编辑系统的设计与搭建

姓 名 罗彬慈

学 院 人工智能学院

专 业 智能科学与技术

班 级 2020219107

学 号 2020212053

指导教师 李佩佩

2024 年      月

# 北京邮电大学

## 本科毕业设计（论文）诚信声明

本人声明所呈交的毕业设计（论文），题目《基于 LLM 的交互式多模态图像编辑系统的设计与搭建》是本人在指导教师的指导下，独立进行研究工作所取得的成果。尽我所知，除了文中特别加以标注和致谢中所罗列的内容以外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得北京邮电大学或其他教育机构的学位或证书而使用过的材料。

申请学位论文与资料若有不实之处，本人承担一切相关责任。

本人签名：\_\_\_\_\_ 日期：\_\_\_\_\_

## 关于论文使用授权的说明

本人完全了解并同意北京邮电大学有关保留、使用学位论文的规定，即：北京邮电大学拥有以下关于学位论文的无偿使用权，其中包括：学校有权保留并向国家有关部门或机构送交学位论文，有权允许学位论文被查阅和借阅；学校可以公布学位论文的全部或部分内容，有权允许采用影印、缩印或其它复制手段保存。汇编学位论文，将学位论文的全部或部分内容编入有关数据库进行检索。（保密的学位论文在解密后遵守此规定）

本人签名：\_\_\_\_\_ 日期：\_\_\_\_\_

导师签名：\_\_\_\_\_ 日期：\_\_\_\_\_

# 北京邮电大学

## 本科毕业设计（论文）任务书

学院	人工智能学院		专业	智能科学与技术	
学生姓名	罗彬慈	学号	2020212053	班级	2020219107
指导教师姓名	李佩佩	所在单位	人工智能学院	职称	副教授
设计(论文)题目	(中文) 基于 LLM 的交互式多模态图像编辑系统的设计与搭建				
	(英文) Design and Construction of Interactive Multimodal Image Editing System Based on LLM				
题目类型	工程实践类 <input type="checkbox"/> 研究设计类 <input checked="" type="checkbox"/> 理论分析类 <input type="checkbox"/> 文献综述类 <input type="checkbox"/> 其他 <input type="checkbox"/>				
题目来源	题目是否来源于科研项目 是 <input checked="" type="checkbox"/> 否 <input type="checkbox"/>				
	科研项目名称:				
	科研项目负责人:				

主要内容：

一：熟悉图像编辑技术和生成模型的相关知识；

支撑指标点：1. 6 2. 1 2. 3 3. 1 3. 2 4. 1 5. 3 10. 1 11. 2

二：分析传统图像编辑模型的限制和挑战；

支撑指标点：1. 6 2. 1 2. 3 3. 1 3. 2 4. 1 5. 3 10. 1 11. 2

三：设计和构建交互式图像编辑系统；

支撑指标点：1. 6 2. 1 2. 3 3. 1 3. 2 4. 1 5. 3 10. 1 11. 2

四：评估系统性能并进行质量控制；

支撑指标点：1. 6 2. 1 2. 3 3. 1 3. 2 4. 1 5. 3 10. 1 11. 2

主要(技术)要求：

内容一：熟悉图像编辑技术和生成模型的相关知识；

1. 6 掌握图像生成与语言大模型基础知识及原理，能够将其和计算机知识与原理、数学与工程方法以及计算求解能力用于分析和解决复杂工程问题，并能够对解决方案进行比较和综合。

4. 1 能够采用科学方法，通过文献研究和应用案例分析等方法，调研和分析领域图像生成与语言大模复杂工程问题的解决方案。

内容二：分析传统图像编辑模型的限制和挑战；

2.1 针对交互式图像编辑领域的复杂工程问题进行问题识别，分析其功能需求与非功能需求，识别其面临的各种制约条件，对任务目标给出需求描述。

2.3 针对已建立的交互式图像编辑领域的复杂工程问题的抽象模型，论证模型的合理性；并通过文献研究，针对改进的可能性进行分析，确定解决方案，获得有效结论。

10.1 能够以撰写报告、设计文稿、口头陈述等方式，针对交互式图像编辑领域复杂工程问题，与业界同行及社会公众进行有效的沟通和交流。

内容三：设计和构建交互式图像编辑系统；

3.1 了解交互式图像编辑系统开发的一般流程，掌握交互式图像编辑系统开发及工程化的基本方法和技术。

3.2 能够针对特定需求，对交互式图像编辑问题进行分解和细化，具有设计/开发功能模块及智能系统的能力。

11.2 能够在多学科环境下，在设计开发交互式图像编辑系统解决方案的过程中，运用工程项目管理与经济决策方法。

内容四：评估系统性能并进行质量控制；

5.3 能够针对交互式图像编辑系统中的具体问题，开发满足特定需求的现代工具，进行仿真和测试，并能够分析其局限性。

主要参考文献：

[1] ACHIAM, Josh, et al. Gpt-4 technical report. arXiv preprint arXiv:2303.08774, 2023.

[2] FLORIDI, Luciano; CHIRIATTI, Massimo. GPT-3: Its nature, scope, limits, and consequences. *Minds and Machines*, 2020, 30: 681-694.

[3] ROMBACH, Robin, et al. High-resolution image synthesis with latent diffusion models. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2022. p. 10684-10695.

[4] HU, Edward J., et al. Lora: Low-rank adaptation of large language models. arXiv preprint arXiv:2106.09685, 2021.

[5] ZHANG, Lvmin; RAO, Anyi; AGRAWALA, Maneesh. Adding conditional control to text-to-image diffusion models. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023. p. 3836-3847.

[6] VAN HUYNH, Nguyen, et al. DeepFake: Deep dueling-based deception strategy to defeat reactive jammers. *IEEE Transactions on Wireless Communications*, 2021, 20.10: 6898-6914.

进度安排：

第1阶段（2023.11.20 – 2024.12.03）：

开始调研基于深度神经网络的图像生成模型，包括对抗生成模型（GANs）、扩散模型等，以了解它们的基本原理和应用领域。

准备开题报告，明确研究目标和方法。

第2阶段（2023.12.04 – 2023.12.17）：

深入研究复现的图像生成模型，理解其优点和限制，探索其在图像编辑任务中的潜在应用。

第3阶段（2023.12.18 – 2023.12.31）：

针对图像编辑任务，提出创新性的问题，明确解决思路。

第4阶段（2024.01.01 – 2024.01.14）：

进行图像编辑模型的实验，收集和分析实验结果，识别性能瓶颈和问题。

研究已有文献和代码，选择合适的图像生成模型进行复现，确保对模型的理解和实现代码能力。

根据图像编辑任务的需求，设计和实施性能提升的方案，可能包括模型改进、数据增强等。

同时开始调研语言大模型，了解其发展现状和应用领域，确保掌握如何使用语言大模型。

第5阶段（2024.02.26 – 2024.03.10）：

结合语言大模型的研究，探讨如何将语言大模型与图像编辑模型相结合，构建交互式图像编辑系统。

第6阶段（2024.03.11 – 2024.03.24）：

进行实验，评估交互式图像编辑系统的性能，进行调整和改进。

第7阶段（2024.03.25 – 2024.04.07）：

文献更新和总结，更新文献调研，将最新研究成果与自己的工作相结合，确保研究与学术前沿保持同步，准备中期报告。

第8阶段（2024.04.08 – 2024.04.19）：

完善研究和论文撰写，完善能量模块的功能，并开始论文撰写，完成研究项目，准备最终论文和答辩

指导教师签字		日期	年   月   日
--------	--	----	-----------

# 北京邮电大学

## 本科生毕业设计（论文）成绩评定表

学生姓名	罗彬慈			学院	人工智能	
学号	2020212053	专业	智能科学与技术		班级	202 021 910 7
论文题目	(中文) 基于 LLM 的交互式多模态图像编辑系统的设计与搭建 (英文) Design and Construction of Interactive Multimodal Image Editing System Based on LLM					
指导教师	指导教师职称				指导教师单位	
中期检查小组评分	中期检查 (满分 30 分) : 26 中期检查小组长签字: 日期: 2024 年 3 月 21 日					
指导教师评分	指导教师成绩评定标准					
	评价内容	具体要求	分值	评分		
				指导教师	复议	
	调研论证	能独立查阅理解文献和从事相关调研；正确翻译外文资料；有分析、综合各种信息、获取新知识及拓展更新知识的能力和自学能力。	3	<b>3</b>		
	方案设计	能独立提出符合需求的可行性研究方案、实验方案、设计方案，独立进行实验（如安装、调试、操作）和研究方案论证。能合理评估系统成本，理解局限性，并考虑社会、健康、安全、法律、文化以及环境等因素的影响。	4	<b>4</b>		
	能力水平	能综合运用所学知识和技能分析解决毕设过程中遇到的实际问题；能正确处理实验数据；能对课题进行理论分析，获得有价值的结论。	3	<b>3</b>		
	学习态度	认真、勤奋、努力、诚实、严格遵守纪律，按期饱满完成规定的任务。	3	<b>3</b>		
	设计（论文）水平	文题相符、综述简练完整，有见解；立论正确，论述充分，结论严谨合理；实验正确，分析处理科学；文字通顺，技术用语准确，符合工程管理规范；设计（论文）有理论价值和应用价值。	5	<b>4</b>		
	文本规范	装订顺序正确，字体字号等与基本规范相符，符号统一，编号齐全，图表完备、整洁、正确。	2	<b>2</b>		
指导教师评分合计 (满分 20 分)			<b>19</b>			

	评语：该毕业设计在内容丰富性和独创性上表现出色，尤其是在实际应用方面，学生成功地将理论知识与实际问题结合，展示了扎实的工程能力和出色的逻辑思维、系统分析能力。该毕业设计工作量饱满，实现效果优秀，论文逻辑清晰，格式规范，很好地达到了预期的研究目标和要求。			
	指导教师签字： 日期：2024 年 5 月 17 日			
复议	<input type="checkbox"/> 是 <input type="checkbox"/> 否 复议评分合计： 复议人签字： 复议日期： 复议有权限修改指导教师评分，选择复议后指导教师评分将由复议评分替换			
本科生毕业设计（论文）答辩成绩评定标准				
答辩小组成绩评定	评价内容	具体要求	分值	评分
	选题	符合专业培养目标，符合社会实际、结合工程实际，难度适度，体现新颖性、综合性。	5	
	设计（论文）质量水平	全面完成任务书中规定的各项要求，文题相符，工作量饱满，写作规范，达到综合训练标准和毕业要求，有理论成果和应用价值，并考虑社会、安全、环境等因素。	20	
	答辩准备	准备充分；有简洁、清晰、美观的演示文稿；准时到场。	5	
	内容陈述	语言表达简洁、流利、清楚、准确，重点突出，逻辑性强，概念清楚，论点正确；实验方法科学，分析归纳合理；结论严谨；对毕业设计（论文）的内容掌握透彻。	15	
	回答问题	回答问题准确、有深度、有理论根据、基本概念清晰。	5	
		答辩小组评分合计（满分 50 分）		
		意见：		
	答辩小组组长签字： 年 月 日			
	答辩小组成员：			
学院意见	最终成绩：百分制                  ;        五分制  院长签章： 学院盖章： 年 月 日			

备注	
----	--

注:

- 1.此表原件一式三份，一份存入学生档案，一份装订到毕业论文中，一份学院教务科留存。
- 2.中期检查成绩、指导教师成绩、最终答辩成绩的评定标准和分数只是基础框架，仅供参考。各学院应围绕毕设目标达成度，结合人才培养目标、专业认证要求等进行具体细化，不能直接用此模板。

# 基于 LLM 的交互式多模态图像编辑系统的设计与搭建

## 摘要

随着深度学习技术在图像处理领域和文本生成领域的迅速发展，多模态交互系统的构建已成为研究的热点。本文介绍了一个基于最新图像生成模型和大语言模型的交互式多模态图像编辑系统的设计与搭建，系统利用了 Stable Diffusion、DALL-E 等图像生成模型和 ChatGPT 系列、ChatGLM2-6B 等大语言模型，通过图形用户界面（GUI）、中间件（middleware）进行图像生成模型和大语言模型的整合，实现了一个既直观又高效的基于文本交互的多模态图像编辑系统。用户可以通过简单的文本指令控制图像编辑过程，系统能够自动解析这些指令并修改图像。同时，本项目通过自动化脚本使用特定图像数据集、图像分割模型和 GPT3.5 Turbo 构建了适用于特定任务的大语言模型微调数据集，使用了 LoRA 方法对 ChatGLM2-6B 模型进行微调，并在微调后在特定任务中获得了近似于 GPT3.5 Turbo 的性能表现。本项目评估了系统在实际应用中的表现，结果显示该系统能够有效地提高图像编辑效率和用户交互体验。

**关键词** 图像编辑 大语言模型 多模态

# **Design and Construction of Interactive Multimodal Image Editing System Based on LLM**

## **ABSTRACT**

With the rapid advancement of deep learning technologies in the fields of image processing and text generation, the construction of multimodal interaction systems has become a focal point of research. This paper presents the design and construction of an interactive multimodal image editing system based on the latest image generation models and large language models. The system utilizes image generation models such as Stable Diffusion and DALL-E, alongside large language models like the ChatGPT series and ChatGLM2-6B. Through a graphical user interface (GUI) and middleware, it integrates these models to create an intuitive and efficient text-based multimodal image editing system. Users can control the image editing process through simple text commands, which the system automatically parses and applies to modify images. Additionally, this project has constructed a fine-tuning dataset for specific tasks using automated scripts with specific image datasets, image segmentation models, and GPT-3.5 Turbo. The LoRA method was used to fine-tune the ChatGLM2-6B model, achieving performance close to that of GPT-3.5 Turbo in specific tasks. The system's performance in practical applications has been evaluated, showing that it can effectively improve image editing efficiency and enhance user interaction experience.

**KEY WORDS** Image Editing Large Language Models Multimodal

# 目 录

<b>第一章 绪论</b>	1
1.1 项目背景与意义	1
1.1.1 项目背景	1
1.1.2 项目意义	1
1.2 国内外研究现状	1
1.2.1 图像编辑	2
1.2.2 大语言模型微调	2
1.2.3 多模态图像编辑方法	3
1.3 项目内容及创新点	4
1.3.1 项目内容	4
1.3.2 项目创新点	5
1.4 论文结构	5
<b>第二章 相关技术研究</b>	6
2.1 扩散模型	6
2.2 基于扩散模型的可控图像生成	7
2.3 大语言模型	7
2.4 系统开发工具	8
2.5 本章小结	9
<b>第三章 基于 LLM 的交互式多模态图像编辑系统的需求分析</b>	10
3.1 系统业务与用户角色分析	10
3.2 系统功能需求分析	10
3.3 本章小结	10
<b>第四章 基于 LLM 的交互式多模态图像编辑系统的设计与实现</b>	11
4.1 GUI 的构建	11
4.1.1 图像自动遮罩与优化	12
4.1.1.1 图像自动遮罩	13
4.1.1.2 图像遮罩性能优化	13
4.1.1.3 对自动生成的遮罩进行优化	14
4.1.2 多模态	15
4.1.2.1 JSON 指令生成	15
4.1.2.2 JSON 指令校验	16
4.1.2.3 多指令处理	17

4.1.2.4 图像模型请求参数生成	17
4.1.3 图像修改建议	17
4.2 Middleware 的构建	18
4.2.1 对多个平台的 API 进行配置和整合	18
4.2.2 使用 Beego 框架提供 API 服务	18
4.3 LLM 的微调	19
4.3.1 LLM 微调数据集生成与性能评估方法	19
4.3.1.1 微调数据集生成	19
4.3.1.2 LLM 指令生成任务性能评估方法	20
4.3.2 ChatGLM2-6B 针对指令生成任务的微调	20
4.3.3 各个 LLM 在本任务下的性能评估	21
4.4 Stable Diffusion 及扩展的使用	23
4.4.1 Stable Diffusion API 的使用	24
4.4.2 ControlNet 的使用及效果	24
4.4.3 Roop 的使用及效果	25
4.5 本章小结	25
<b>第五章 系统实现效果与使用</b>	<b>26</b>
5.1 系统实现效果	26
5.2 系统使用方法	26
5.2.1 系统部署	26
5.2.2 GUI 使用说明	28
5.3 本章小结	29
<b>第六章 项目管理与维护</b>	<b>30</b>
6.1 代码管理	30
6.2 自动化测试	30
6.3 持续集成与持续部署	31
6.4 本章小结	32
<b>第七章 总结及未来展望</b>	<b>33</b>
7.1 总结	33
7.2 未来展望	34
<b>参考文献</b>	
<b>致    谢</b>	
<b>开  题  报  告</b>	
<b>中  期  检  查  表</b>	
<b>教师指导毕业设计(论文)记录表</b>	

# 第一章 绪论

## 1.1 项目背景与意义

### 1.1.1 项目背景

随着技术的迅速发展，图像生成编辑在媒体娱乐、数字营销等领域等多个行业中发挥越来越重要的作用，然而传统的图像编辑技术在交互性和生成图像质量上仍面临许多挑战。传统图像编辑工具往往依赖于专业的技术知识和复杂的操作界面，交互性通常较差，不能很好地根据用户的具体需求进行灵活调整和响应，对于普通用户来说门槛较高，用户需要花费大量时间学习如何使用这些工具，限制了工具的普及性和易用性。

为了改善图像编辑的交互性和灵活性，人们正在利用深度学习技术来改善图像编辑系统。目前的研究显示扩散模型已经在图像生成领域具有巨大的潜力，其能够学习大量图像数据，自动提取复杂的特征，并生成高质量的图像。若将其运用于图像编辑，或许可以改善图像编辑的质量。对于交互性，若将最新的 GPT4Turbo 等性能优异的大语言模型融入到图像编辑系统中，或许可以进一步提升系统的交互性，实现更加自然和直观的交互。

通过将图像生成模型与大语言模型结合，可以创建一个更加灵活且易用性强的图像编辑系统，该系统不仅能够提供更加直观的编辑界面，降低用户的操作难度，还能根据用户的描述自动生成或修改图像，极大地提升生成图像的质量和编辑效率。用户可以通过简单的语言指令，如“增加图片亮度”或“改变背景为海滩”，直接与编辑系统交互，系统能够理解这些指令并即时作出响应。

通过整合深度学习和语言模型技术，我们有望构建出一个全新的交互式图像编辑系统，其不仅能够提供更高质量的图像编辑结果，更能够提供给用户更加自然的交互方式，为该领域的专业人士和普通用户都带好的图像编辑体验。

### 1.1.2 项目意义

通过融合先进的图像生成技术与大语言模型，本项目希望能搭建一个基于 LLM 的交互式图像编辑系统，提供更为易用、精准且高效的图像编辑工具，提升媒体娱乐、数字营销以及智能医疗等多个行业的图像处理能力。通过实现更加智能化和用户友好型的编辑系统，为广大用户带来前所未有的图像编辑体验。这样的研究与开发，为图像编辑技术的未来提供了一种可能性和一条新的探索和实践路径。

## 1.2 国内外研究现状

Multimodal image synthesis and editing: A survey<sup>[1]</sup> 这篇论文对多模态图像合成和编辑的进展进行了综述，讨论如何有效地结合多种模态信息来创造和编辑图像。该论文提出了一种分类体系，根据数据模态和模型架构进行分类，并详细介绍了多模态图像合成

和编辑的各种方法。该论文还讨论了基准数据集、评价指标以及当前研究中的挑战，并提出了未来的研究方向，强调在图像合成和编辑任务中引入跨模态指导的重要性和潜力。

### 1.2.1 图像编辑

ImageBART<sup>[2]</sup> 是一种基于自回归变换的图像生成和编辑模型，灵感来源于自然语言处理中的 BART(Bidirectional and Auto-Regressive Transformers)<sup>[3]</sup>，其结合了自回归和编码-解码架构，用于高效地生成和编辑图像。EditGAN<sup>[4]</sup> 提出了一种高精度的语义图像编辑方法，允许用户通过修改图像的详细部分分割掩模来编辑图像，其基于生成对抗网络(GAN)构建，只需要少量标记样本即可进行训练，实现了高效的编辑。Generating images from captions with attention<sup>[5]</sup> 介绍了一个能从自然语言描述中生成图像的模型，这个模型在关注描述中的相关词汇的同时利用深度循环注意力编写器(DRAW)迭代地在画布上绘制图像。Object-based image editing<sup>[6]</sup> 介绍了基于对象的图像编辑技术，允许用户直接在对象层面而不是像素层面进行编辑。Faceshop: Deep sketch-based face image editing<sup>[7]</sup> 提供了一种利用深度学习来进行面部图像编辑的方法。Image-based modeling and photo editing<sup>[8]</sup> 探讨了基于图像的建模与照片编辑技术，特别强调了分层编辑和实体分离技术的应用。Invertible conditional gans for image editing<sup>[9]</sup> 探索了可逆条件 GANs 在图像编辑中的应用，这种模型结合了编码器和 cGAN，可对真实图像进行精确修改。In-domain gan inversion for real image editing<sup>[10]</sup> 研究了在领域 GAN 反转技术在真实图像编辑中的应用，通过 GAN 学习特定图像域的编辑操作。Poisson image editing<sup>[11]</sup> 探讨了泊松图像编辑技术，该技术能够无缝地将图像区域融合和编辑，并提供高质量的编辑效果。Image editing in the contour domain<sup>[12]</sup> 提出了一种在轮廓域进行图像编辑的方法，这种方法直接在轮廓层面而不是像素层面进行操作。Diffusion maps for edge-aware image editing<sup>[13]</sup> 使用扩散映射技术进行边缘感知图像编辑以在保持边缘清晰的同时平滑颜色和纹理。

### 1.2.2 大语言模型微调

Parameter-efficient fine-tuning of large-scale pre-trained language models<sup>[14]</sup> 介绍了 Delta-tuning 技术，它在不改变预训练大模型(PLMs)架构的基础上，通过微调少量参数来适应新任务。其主要包括以下方法：增加式、指定式和重新参数化方法，以及在变换器层添加适配器模块的实际应用，这些方法比全参数微调更节省计算资源，提高了模型的适应性和效率。How fine can fine-tuning be? learning efficient language models<sup>[15]</sup> 分析了微调过程的效率，对在保持性能的同时减少计算成本的方法进行探索。# InsTag: Instruction Tagging for Analyzing Supervised Fine-tuning of Large Language Models<sup>[16]</sup> 使用指令标签分析监督式微调过程，以提高大型语言模型在特定任务上的性能。Scaling federated learning for fine-tuning of large language models<sup>[17]</sup> 应用联邦学习方法来微调大型语言模型，以提高模型在多个数据源上的通用性和隐私保护。Longlora: Efficient fine-tuning of

long-context large language models<sup>[18]</sup> 针对长上下文的大型语言模型进行高效微调，以提升模型对长距离依赖信息的处理能力。Fine-tuning pre-trained language models effectively by optimizing subnetworks adaptively<sup>[19]</sup> 通过自适应优化子网络来有效微调预训练语言模型，以提高模型的灵活性和适应新任务的能力。Llm-adapters: An adapter family for parameter-efficient fine-tuning of large language models<sup>[20]</sup> 提出一种用于大型语言模型参数高效微调的适配器家族，以减少模型调整过程中的资源消耗。Language models are few-shot learners<sup>[21]</sup> 探讨语言模型在小样本学习中的表现，并分析其微调和适应新任务的能力。Fine-tuning language models with just forward passes<sup>[22]</sup> 提出一种仅通过前向传播进行微调的新方法，以简化语言模型的调整过程。

### 1.2.3 多模态图像编辑方法

PixelTone: A Multimodal Interface for Image Editing<sup>[23]</sup> 介绍了一种结合语音和直接操作的多模态照片编辑界面，旨在简化移动设备上的图像编辑任务，通过自然语言和草图来定位图像中的具体更改区域。研究还开发了一个定制的自然语言解释器，将用户的语言指令映射到具体的图像处理操作上。通过用户研究，验证了接口的有效性，展示了其在简化编辑流程和提高用户互动体验方面的潜力。DiffusionCLIP<sup>[24]</sup> 是一种结合了扩散模型和 CLIP<sup>[25]</sup> 模型的新型图像编辑方法。这种方法利用了扩散模型的生成能力和 CLIP 的语义理解能力，以实现通过文本描述来精确控制图像的内容和属性的编辑。Language-based image editing with recurrent attentive models<sup>[26]</sup> 提出一种基于语言的图像编辑方法，通过递归注意模型允许用户用自然语言描述来指导图像编辑过程。Imagic: Text-based real image editing with diffusion models<sup>[27]</sup> 利用扩散模型和文本描述来进行真实图像的编辑，并允许细粒度控制和高度个性化的编辑。Sequential attention GAN for interactive image editing<sup>[28]</sup> 介绍了一个用于交互式图像编辑的顺序注意力 GAN 模型，允许用户通过多轮对话逐步指导图像编辑。Towards automatic image editing: Learning to see another you<sup>[29]</sup> 研究了自动图像编辑的可能性，通过机器学习方法让系统能够根据用户的需求生成或修改图像。Instructpix2pix: Learning to follow image editing instructions<sup>[30]</sup> 使用 pix2pix 模型学习遵循图像编辑指令，使得模型能够根据文字描述自动进行图像编辑。Blended diffusion for text-driven editing of natural images<sup>[31]</sup> 使用自然语言界面和数据驱动的图像生成技术进行文本驱动的图像编辑。Tigan: Text-based interactive image generation and manipulation<sup>[32]</sup> 提出了一种基于文本的交互式图像生成和操作框架。Shape-aware text-driven layered video editing<sup>[33]</sup> 对基于形状感知的文本驱动分层视频编辑方法进行探索。Prompt tuning inversion for text-driven image editing using diffusion models<sup>[34]</sup> 使用扩散模型，为基于文本的图像编辑提供一种新的反演方法。Lightweight text-driven image editing with disentangled content and attributes<sup>[35]</sup> 提供了通过解耦内容和属性以进行轻量级的基于文本驱动的图像编辑的方法。

## 1.3 项目内容及创新点

### 1.3.1 项目内容

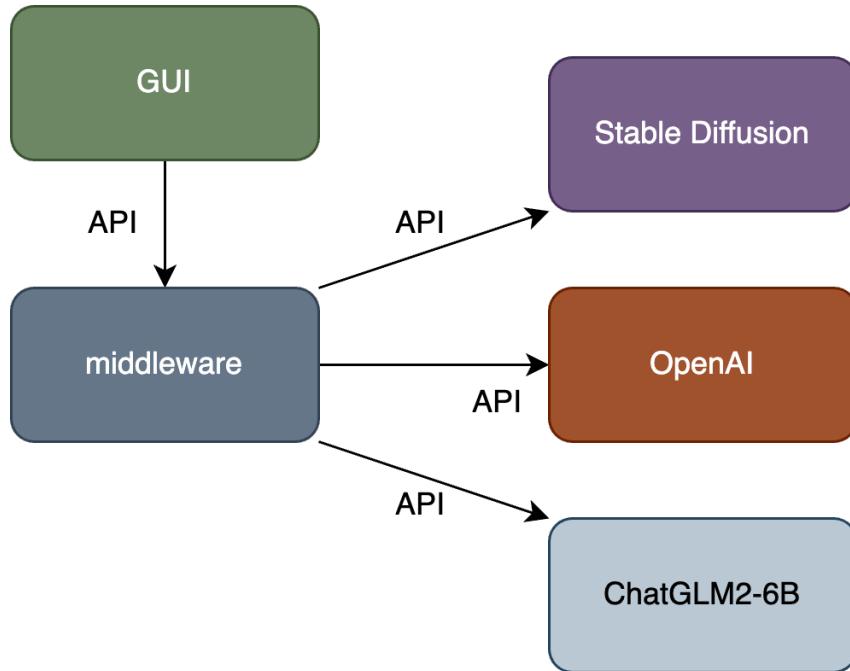


图 1-1

该项目是一个多模态交互式图像编辑系统，它主要实现了 GUI、Middleware、以及对 Stable Diffusion 和 ChatGLM2-6B 模型的修改与适配。在整体结构上，GUI、Middleware、以及模型修改之间的相互关系和数据流向见图 1-1。

项目基于 Stable Diffusion 的开源项目 stable-diffusion-webui 进行了扩展，增强了其功能。系统可以调用不同的 Stable Diffusion 模型并结合多个扩展后的功能来对图像进行精细的修改和调整。这一功能的实现大大增强了系统对图像处理的灵活性和多样性。

项目还包括了对 ChatGLM2-6B 模型的微调。通过利用专门为本项目的需求生成的微调数据集进行微调，进一步提升了语言模型处理特定任务的能力和准确性，且能够利用 API 调用这些经过微调的 ChatGLM2-6B 模型。

通过调用 OpenAI 的 API，本项目实现了多项功能：利用 GPT4V 生成关于图像修改的建议，使用 GPT3.5Turbo 来辅助生成用于微调大语言模型的数据集以及图像修改指令，以及在 Stable Diffusion 模型不可用时，使用 DALL-E2 作为替代模型来进行图像修改。这些功能为基于 LLM 的交互式图像编辑系统提供了强大的工具。

在 GUI 方面，本系统主要通过 Python 语言实现，构建了一个直观且用户友好的交互界面。该界面简洁易用，通过 API 调用向 Middleware 发出请求，有效减轻了用户在执行计算密集型任务时对硬件资源的需求，从而降低了用户侧的使用门槛。

Middleware 部分使用 Golang 语言构建了一个后端服务。这个服务不仅接入了 Stable Diffusion、ChatGLM2-6B、OpenAI 的 API，还将这些 API 进行了有效的整合，向 GUI 提

供了一致风格的 API 接口。这种设计不仅提升了 GUI 调用多方 API 的便利性，也通过统一的配置和管理，极大地增强了系统的可维护性和稳定性。

通过整合深度学习和大语言模型技术，本项目不仅能够提供更高质量的图像编辑结果，还能为用户提供更加自然的交互方式，极大地提升了图像编辑的效率和体验。此外，系统设计中还考虑了扩展性和未来技术的整合，预留了接口以适应未来可能的升级，以适应快速发展的技术需求。

### 1.3.2 项目创新点

本项目的方案设计充分考虑了技术实现的可行性与用户操作的便捷性，力求在满足复杂功能需求的同时，保证系统的易用性和稳定性。本项目将复杂的模型调用参数抽象为用户可理解的模版与设置，建立了一个基于 JSON 的指令机制打通了大语言模型与图像生成模型，同时对多种主流大语言模型和图像编辑模型进行适配预留有接口，可随着大语言模型和图像生成模型技术的发展快速迭代。通过系统化的设计和技术的整合，本项目实现了一个高效且用户友好的多模态交互式图像编辑系统。

## 1.4 论文结构

本论文共分为七个章节，每个章节的具体内容如下：

第一章，绪论。本章主要论述了基于 LLM 的交互式多模态图像编辑系统的项目背景与意义，总结了相关的国内外研究现状，简述了项目内容并列举了项目创新点。

第二章，相关技术研究。本章主要介绍了本项目所使用到的相关技术，包括扩散模型、基于扩散模型的可控图像生成、大语言模型、系统开发工具。

第三章，基于 LLM 的交互式多模态图像编辑系统的需求分析。本章主要对系统业务与用户角色进行分析，结合本项目所使用技术的特点和传统图像编辑系统的痛点，分析了系统功能需求。

第四章，基于 LLM 的交互式多模态图像编辑系统的设计与实现。本章主要论述系统的设计与实现方式，包括 GUI 与 Middleware 的构建、多模态的实现、LLM 的微调、StableDiffusion 及扩展的使用。

第五章，系统实现效果与使用。本章主要展示了系统实现效果，并对系统使用方法进行说明。

第六章，项目管理与维护。本章主要论述了项目所使用的管理与维护方法，包括代码管理、自动化测试、持续集成与持续部署。

第七章，总结及未来展望。本章对全文和开发工作进行总结与归纳，并提出基于 LLM 的交互式多模态图像编辑系统的未来展望。

## 第二章 相关技术研究

### 2.1 扩散模型

扩散模型是近年来发展起来的一种新型生成模型，与传统的生成对抗网络（GANs）和变分自编码器（VAEs）相比，扩散模型在生成图像的质量和多样性方面展现出了卓越的性能，其基本原理是模拟从高质量数据分布到高熵噪声分布的逐步转变过程，然后再逆向这一过程以生成新的数据。扩散模型的关键优势在于其生成的图像质量较高且自然，而且在训练过程中相对稳定，不易出现生成对抗网络中常见的模式崩溃问题，其基本原理如下：

前向过程（Forward Process）也称为扩散过程逐步将原始数据  $x_0$  加入高斯噪声，最终转化为纯噪声  $x_t$ 。此过程可以表示为算法 1：

---

#### 算法 1 扩散模型的前向过程

---

- 1: 初始化  $x_0 \sim q(x_0)$
- 2: **for**  $t = 1$  到  $T$  **do**
- 3:      $x_t \sim q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1 - \beta_t}x_{t-1}, \beta_t\mathbf{I})$
- 4: **end for**

---

反向过程（Reverse Process）是前向过程的逆过程，其能通过通过训练的参数化模型  $\epsilon_\theta(x_t, t)$  从纯噪声状态  $x_t$  逐步重构出原始数据  $x_0$ 。此过程可以表示为算法 2：

---

#### 算法 2 扩散模型的反向过程

---

- 1: 初始化  $x_T \sim \mathcal{N}(0, \mathbf{I})$
- 2: **for**  $t = T$  到  $1$  **do**
- 3:     如果  $t > 1$ ，则采样  $z \sim \mathcal{N}(0, \mathbf{I})$ ，否则  $z = 0$
- 4:      $x_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left( x_t - \frac{\beta_t}{\sqrt{1 - \bar{\alpha}_t}} \epsilon_\theta(x_t, t) \right) + \sigma_t z$
- 5: **end for**
- 6: **return**  $x_0$

---

扩散模型的优化最小化真实噪声和模型估计噪声之间的差异，其过程可以表示为算法 3：

---

#### 算法 3 扩散模型的优化过程

---

- 1: **repeat**
- 2:     采样  $x_0 \sim q(x_0)$ ,  $t \sim \text{Uniform}\{1, \dots, T\}$ ,  $\epsilon \sim \mathcal{N}(0, \mathbf{I})$
- 3:      $\tilde{x}_t = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon$
- 4:     对  $\nabla_\theta \|\epsilon - \epsilon_\theta(\tilde{x}_t, t)\|^2$  进行梯度下降
- 5: **until** 收敛

---

Stable Diffusion<sup>[36]</sup> 是一种基于扩散模型的深度学习图像生成模型，它能够根据文本描述生成高质量的图像。该模型采用了条件生成技术，允许用户通过文本指令来引导噪

声逆向还原从而生成高质量的图片，在艺术创作、媒体娱乐、广告和数字营销等多个领域具有广泛的应用。

由于本项目对于图像生成模型的要求较高且需求复杂，为了便于结合 Stable Diffusion 模型和其他前沿研究成果及开源社区项目，本项目在构建 Stable Diffusion 模块时以开源项目 stable-diffusion-webui<sup>1</sup>为基础，结合 sd-webui-controlnet 和 sd-webui-roop 等扩展，通过 API 为 Middleware 提供服务。当 Stable Diffusion 模型不可用时，系统则会调用 OpenAI 的 DALL-E 2 模型。DALL-E 2 是一个先进的图像生成模型，可以根据用户提供的文本描述生成详细、高质量的图像，其核心优势在于其创造力和多样性，能够在遵循描述的同时，创造出独特和富有创意的视觉内容。

## 2.2 基于扩散模型的可控图像生成

ControlNet<sup>[37]</sup> 是一种新的神经网络架构，用于在大型预训练的文本到图像扩散模型中添加空间条件控制。ControlNet 的核心是利用预训练模型的深层和健壮的编码层作为强大的支撑，学习多种条件控制。该网络通过零初始化的卷积层（zero convolutions）连接，这些层从零开始逐步增长参数，确保训练初期不会引入有害的噪声。通过使用 ControlNet，用户可以更精细地控制图像生成过程，使生成的图像更贴近用户的具体需求，尤其是在空间布局和细节表达上。这种方法能够有效地减少试错循环，提高图像生成的效率和质量。其与 Stable Diffusion 结合的方式如图 2-1 所示（图片摘自论文 Adding Conditional Control to Text-to-Image Diffusion Models<sup>[37]</sup>）。

## 2.3 大语言模型

近些年大语言模型发展异常迅猛，其通过学习大量文本数据，能够出色地完成生成文本、回答问题、翻译语言等任务。随着算力的提升和语料的增加，大语言模型已经取得了显著的进步，并在多个应用场景中展现出了强大的能力。目前大语言模型主要使用 Transformers<sup>[38]</sup> 架构，其能够有效处理长距离依赖问题。GPT (Generative Pre-trained Transformer)、BERT (Bidirectional Encoder Representations from Transformers) 等主流大语言模型，通过大规模的语料进行预训练，对语言的深层次结构和语义的理解能力有了显著的提升，在应用方面展现出广泛的适用性。在大语言模型中，最著名的便是 OpenAI 的 GPT 系列：GPT-3.5 Turbo 是 OpenAI 开发的一款先进的自然语言处理模型，属于 GPT-3 系列的增强版本，在处理大量文本和生成文本方面表现出色；GPT-4<sup>[39]</sup> 是 GPT-3 的后续版本，代表了目前最新一代的大语言模型技术，在模型结构和训练数据量上进行了大幅扩展，能够更准确地理解和生成复杂的文本，在理解上下文、维持一致性以及生成更自然的语言方面具有显著优势；GPT-4V 是 GPT-4 的一个特殊版本，优化了图像标注、视觉问答等视觉任务，结合文本和视觉处理能力，能够更好地理解和生成与

<sup>1</sup><https://github.com/AUTOMATIC1111/stable-diffusion-webui>

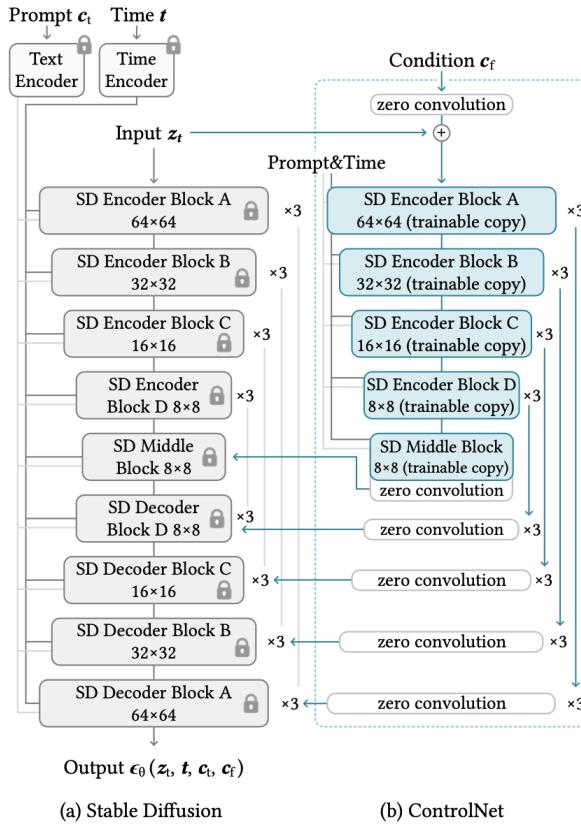


图 2-1

图像相关的文本内容。

ChatGLM2-6B 是由清华大学开发的第二代开源双语(中英)对话模型, 基于 ChatGLM-6B 进行迭代并提升了性能, 这款模型经过大规模预训练, 实现了显著的性能提升, 并在多个数据集上表现出色。ChatGLM2-6B 支持更长的对话上下文, 并提高了推理速度和降低了显存占用, 使得即使在资源有限的环境下也能有效运行。本项目使用 ChatGLM2-6B 模型, 结合开源项目 LLaMA-Factory<sup>1</sup>, 利用本项目提供的数据自动生成功能所生成的数据集, 使用 LoRA<sup>[40]</sup> 方法对模型进行微调以在本项目所需的任务中获得更佳的表现。微调后的模型通过 fastapi 提供 API 服务。

LoRA 通过在每个 Transformer 层中注入可训练的低秩矩阵对权重进行更新, 而不是更新整个权重矩阵。这种方法不仅减少了模型的存储和计算成本, 还保持了与完全微调相当的模型性能。其算法可概括为算法 4。

## 2.4 系统开发工具

Gradio 是一个开源的 Python 库, 旨在简化为机器学习模型创建自定义用户界面的过程。本项目使用 Gradio 框架构建了 GUI 组件以提供简单易用的交互界面。

beego 是一个使用 Go 语言开发的开源 Web 框架, 它支持快速开发各种应用程序,

<sup>1</sup><https://github.com/hiyouga/LLaMA-Factory>

---

**算法 4 LoRA: 低秩适应大型语言模型**

---

输入: 预训练模型权重  $W_0 \in \mathbb{R}^{d \times k}$ , 输入  $x$ , 低秩  $r$

输出: 适应后的输出

- 1: 初始化低秩矩阵  $A \in \mathbb{R}^{d \times r}$  和  $B \in \mathbb{R}^{r \times k}$
  - 2: 设定更新权重  $\Delta W = BA$
  - 3: 冻结预训练权重  $W_0$
  - 4: **for** 每一层 Transformer **do**
  - 5:     使用  $W = W_0 + \Delta W$  进行前向传播
  - 6:     优化  $A$  和  $B$  来最小化损失函数
  - 7: **end for**
  - 8: 输出调整后的模型结果
- 

包括 API、Web 应用和后端服务。该框架设计灵感来源于 Tornado、Sinatra 和 Flask，结合了 Go 的接口 (interface) 和结构体嵌入 (struct embedding) 等特性，提供了高效的性能和简便的操作。本项目使用 beego 框架构建了 Middleware 组件以提供多个 API 服务，以便 GUI 可以高效地使用各种模型和工具。

## 2.5 本章小结

在本章中，我们深入探讨了与多模态图像编辑系统相关的关键技术，首先详细介绍了扩散模型的工作原理及其在图像生成中的应用，接着讨论了大语言模型的相关信息与 LoRA 微调技术，最后概述了系统开发中使用的主要工具和方法。本章不仅为理解系统的技术基础提供了理论支撑，也为后续章节中系统设计和实现的详细讨论奠定了基础。

## 第三章 基于 LLM 的交互式多模态图像编辑系统的需求分析

### 3.1 系统业务与用户角色分析

本项目的目标是构建一个基于 LLM 的交互式多模态图像编辑系统，结合大语言模型和图像生成技术，提供一个友好的图形用户界面，支持用户通过自然语言指令进行复杂的图像编辑操作，使非技术用户也能轻松使用复杂的图像编辑功能。系统需能够理解和解析用户的自然语言指令，将其转换为具体的图像编辑任务，并利用最新的图像生成模型如 Stable Diffusion 根据由用户的指令生成的相关参数修改图像。

本项目的目标用户群体包括设计师、营销专业人员、教育工作者和普通消费者。设计师可以利用系统快速实现创意构思，改善设计流程的效率。营销专业人员可以使用该系统快速调整广告图像，以适应不同的市场需求。教育工作者可以使用此系统来创建或修改教学资料中的图像，使教学内容更加生动有趣。普通消费者则可以利用系统提供的易于使用的平台，探索个人图像编辑和创意表达。

### 3.2 系统功能需求分析

项目旨在整合 Stable Diffusion 和大语言模型等最新的深度学习模型以实现通过自然语言指令实现高质量的图像编辑。图形用户界面 (GUI) 设计应简洁直观，包含预览功能，提升用户体验。中间件 (Middleware) 需高效处理 GUI 请求，并向模型 API 转发，确保系统稳定性和响应速度。系统应能高效处理复杂的图像编辑任务，并考虑多种使用场景和用户需求，以保证广泛应用和良好用户体验。

### 3.3 本章小结

本章主要对基于 LLM 的交互式多模态图像编辑系统进行了需求分析。项目旨在结合大语言模型和图像生成技术提供一个高效、直观的图像编辑系统，使用户能够轻松使用复杂的图像编辑功能。项目的目标用户群体包括设计师、营销专业人员、教育工作者和普通消费者，需支持用户通过自然语言指令利用图像生成模型进行高质量的图像编辑操作。系统需要整合多个深度学习模型、设计直观的 GUI 界面、实现高效处理请求的中间件，并考虑多种使用场景和用户需求，以提供良好的用户体验。

## 第四章 基于 LLM 的交互式多模态图像编辑系统的设计与实现

在本项目中，图形用户界面（GUI）作为用户与系统交互的前端界面扮演了至关重要的角色，不仅需要具备直观操作的特性，还应支持复杂的自定义图像处理功能，因此本项目将简洁且直观考虑为 GUI 设计中的重点，通过图形化元素如按钮、图标和菜单等，允许用户以简单的点击操作来进行交互。GUI 还集成了一些预览功能，用户可以查看遮罩效果，有效提升用户体验和操作的精确性。

Middleware 负责处理来自 GUI 的请求，在本系统中起到了桥梁的作用。Middleware 采用了高效的 Golang 语言结合 beego 框架进行构建以保证系统的响应速度和稳定性，将来自 GUI 的请求进行处理并向对应的部署在云平台或 OpenAI 的模型调用 API 转发。它整合了包括 Stable Diffusion、ChatGLM2-6B 及 OpenAI 提供的 API 等多种服务，通过提供统一风格的 API 接口，极大地简化了 GUI 与模型之间的交互复杂度，不仅提高了开发效率，也便于系统的后期维护和升级。

图像编辑和文本交互方面所用到的图像生成模型和大语言模型是本项目的核心技术。项目中采用了最新的扩散模型技术——Stable Diffusion，其通过学习大量的图像数据，能够生成高质量的图像内容。项目还引入了大语言模型（如 GPT 系列和 ChatGLM2-6B）来处理和理解用户的自然语言指令，实现更加智能的图像编辑功能。用户可以使用简单的语言描述如“将背景更换为海滩”来进行图像编辑，系统能够将用户输入自动解析为特定指令并进行相应的图像编辑。

通过整合 GUI、Middleware、图像生成模型和大语言模型，本项目能够接受用户的文本输入并进行相应的复杂图像编辑。系统的设计考虑到了多种使用场景和用户的不同需求，确保了广泛的应用性和良好的用户体验。

### 4.1 GUI 的构建

图形用户界面是现代软件项目中不可或缺的组成部分，它极大地提升了应用程序的可访问性和用户体验。GUI 通过可视化元素如按钮、图标和菜单等，允许用户以直观的方式与系统进行交互，简化了操作过程并降低了用户的使用门槛。本项目所构建的 GUI 为用户提供了一个交互方式简单且功能强大的图形用户界面，通过图形化的方式展示信息和选择，使得用户能够通过简单的点击或触摸来执行命令或更改设置。GUI 虽然承担计算任务较少，但却是承载本项目结构与逻辑的关键部分。通过使用符合规则的指令作为中枢，GUI 打通了大语言模型和图像生成模型之间的壁垒，使基于 LLM 的创新交互式图像编辑系统成为可能。GUI 的模块构成如表 4-1：

用户首先上传需要修改的图片，然后可在 Chat 模块中选择不同的大语言模型进行交互并得到相应的指令，最后在 Operation Board 模块中选择指令执行或一键全部执行。如果对自动生成的遮罩不满意，可在 Edit Image 中对遮罩进行修改。

在 Auto 模块中，用户可通过选择多张图片批量生成满足微调大语言模型微调所需的数据。其会循环地从给定的图片集中随机选择图片继续分割，将分割后的结果和特定

表 4-1 GUI 模块

模块	描述
BaseImage	接受上传的原始图片并预览
EditedImage	预览修改后的图片
Operation Board	执行指令
Settings	对系统进行设置
Chat	与大语言模型交互的聊天界面
Edit Image	对图像进行自定义遮罩和换脸等操作
Auto	执行自动化操作
Manual	系统使用说明

的 prompt 通过 GPT3.5Turbo 生成对应的修改建议，再将分割的结果、生成的建议通过 GPT3.5Turbo 生成指令。

#### 4.1.1 图像自动遮罩与优化

由于本项目需要提供对图像进行部分修改的功能，所以需要在使用图像生成模型进行图像编辑时需要提供一个遮罩以明确需要修改的部分和不需要修改的部分。为了自动生成符合要求的遮罩，本项目借助图像分割和大语言模型的辅助，可通过两种方式生成自动遮罩：基于关键词对自动生成遮罩和基于已给出的点自动填充生成遮罩。两种方法都会首先使用图像分割模型对图像进行分割（如图 4-1），然后根据给出的要求对相应的部分进行遮罩生成原始的遮罩。受制于图像分割模型在边缘上的表现并不理想，需要对特定的分割部分进行处理以提高遮罩的质量，因此最后会通过本项目设计的优化算法生成最终的遮罩。



图 4-1 图像分割结果：(a)原始图像，(b)分割结果

#### 4.1.1.1 图像自动遮罩

基于关键词自动生成遮罩的方法会根据关键词和图像分割结果生成自动原始的遮罩，该功能会遍历每个给出的关键词，若关键词与分割结果之一吻合，则会对相应的分割区域进行遮罩，生成原始的遮罩（如图 4-2）。

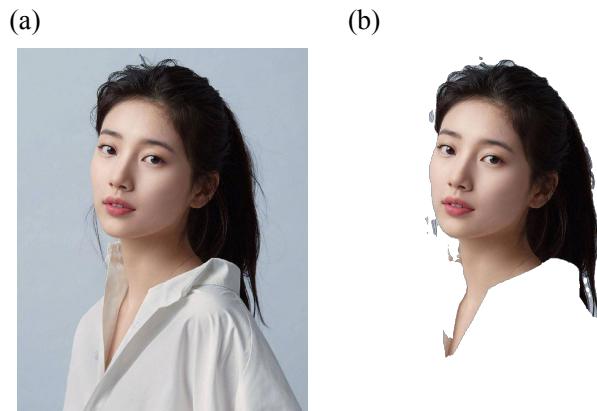


图 4-2 关键词自动生成遮罩结果：(a)原始图像，(b)keywords=[’Background’, ’Upper-clothes’, ’Dress’, ’Right-arm’] 得到的遮罩

基于已给出的点自动填充生成遮罩的方法会根据在图片中标记的点和图像分割结果生成自动原始的遮罩，该功能会遍历每个给出的点，将该点所在的部分全部进行遮罩，最后生成原始的遮罩（如图 4-3）。



图 4-3 基于已给出的点自动填充生成遮罩：(a)标记后的图像，(b)生成的遮罩

#### 4.1.1.2 图像遮罩性能优化

LRU(Least Recently Used)缓存是一种常用的缓存淘汰算法，用于在有限的缓存空间中管理数据。它的核心思想是优先淘汰最长时间未被使用的数据项。functools.lru\_cache 是 python 标准库中 functools 模块提供的一个装饰器，它实现了 LRU 缓存机制。该装饰器可以非常方便地被添加到任何想要进行缓存的参数可哈希的函数上，自动地保存最近

执行的函数调用结果并在后续相同的调用中直接返回缓存的结果，避免重复计算的开销。

由于在本项目中图像遮罩存在一张图片多次调用的特点，本项目使用了 LRU 缓存实现性能优化。由于 python 中 `PIL.Image.Image` 对象不可哈希，缓存分割结果时将图像转为 base64 字符串进行映射。

#### 4.1.1.3 对自动生成的遮罩进行优化

由于分割模型性能的限制，生成的原始遮罩可能在某些细节上表现不佳而影响图像编辑模型的结果，因此设计了一个算法对自动生成的遮罩进行优化。算法 5 可以根据配置文件的设置，对特定的未被遮罩的部分在遮罩的边缘进行收缩。

---

##### 算法 5 遮罩优化算法

---

**输入：** 原始遮罩 *OriginMask*，图像分割结果 *SegmentResult*，配置文件 *Config*

**输出：** 优化后的遮罩 *OptimizedMask*

- 1: 获取遮罩与非遮罩的描边得到像素 *EdgePixels*
  - 2: 从配置文件和图像分割结果获取 *ConfigPixels*
  - 3: 仅保留出现在 *EdgePixels* 中的 *ConfigPixels*
  - 4: **for** *pixel* in *ConfigPixels* **do**
  - 5:     Apply MinFilter Kernel(in *Config*) in *OriginMask*[*pixel*]
  - 6: **end for**
  - 7: 得到优化后的遮罩 *OptimizedMask*
- 

由于该算法仅会对遮罩边缘上的像素进行卷积且在设计时充分考虑到了内存中像素的存储顺序的原因，虽然需要复杂的处理过程，但经过多次的迭代后算法的时间复杂度降低至  $O(mnr)$  ( $m,n$  表示图片的长宽， $r$  表示设置的优化强度)。算法实现的效果如图 4-4 所示，可见在发丝附近遮罩的质量得到了明显的改善。

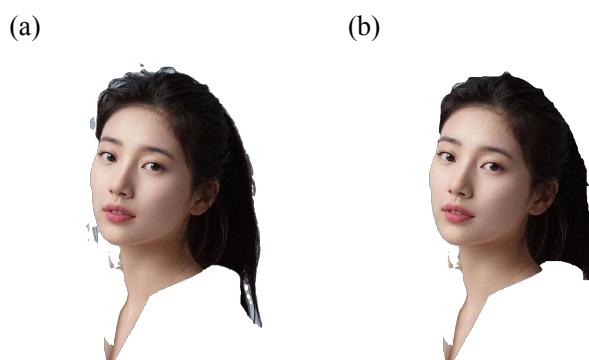


图 4-4 遮罩优化结果：(a)原始遮罩，(b)算法优化后的遮罩

### 4.1.2 多模态

如何打通大语言模型和图像生成模型是本项目的关键。本项目通过特定的 prompt 和图像分割结果，使用大语言模型生成 JSON 格式的指令并校验，并支持多轮对话。用户可有选择性地执行生成的指令或执行全部指令。系统首先会按照给定的规则对指令进行预处理和排序，然后通过指令生成请求参数来调用图像生成模型。多模态任务的实现方式如图 4-5。

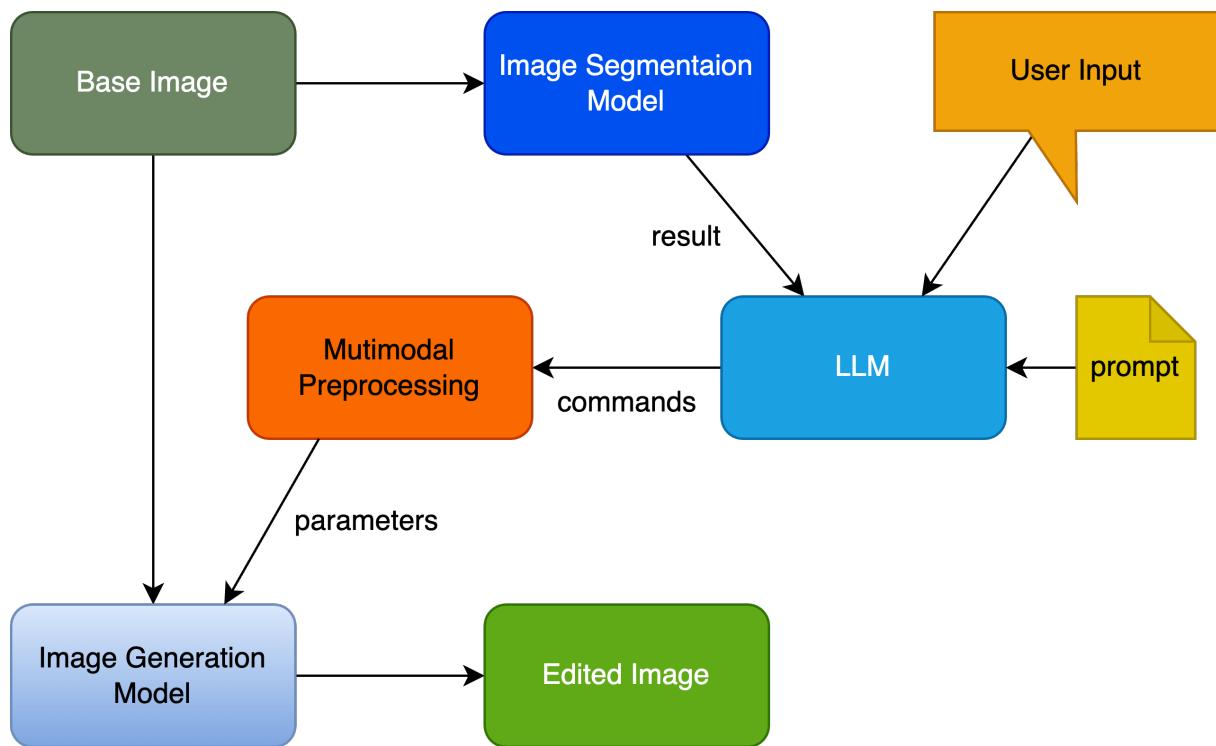


图 4-5

#### 4.1.2.1 JSON 指令生成

JSON (JavaScript Object Notation) 是一种轻量级的数据交换格式，易于人阅读和编写，同时也易于机器解析和生成。它基于 JavaScript 的一个子集，但独立于语言，被广泛应用于许多编程语言中。JSON 主要用于网络应用间浏览器与服务器之间的数据传输。在 JSON 中，数据以键值对的形式存在，可以表示数组、布尔值、数字、对象或字符串。由于其简洁性和易于交互的特性，JSON 已成为 Web 应用中数据交换的主流技术。由于 JSON 应用范围广且大语言模型 JSON 处理能力较强，本项目采用此格式来承载大语言模型和图像生成模型的联系。

通过特定的 prompt 和图像分割结果以及用户输入的修改意图，本项目可使用 GPT3.5Turbo、GPT4Turbo、微调后的 ChatGLM2-6B 生成 JSON 指令。例：当图像分割结果为 Background, Hair, Upper-clothes, Dress, Face, Right-arm，用户输入为“将背景更换为蓝天白云，将衣服更改为白色的 T-shirt”时，生成的 JSON 指令如代码 4-1 所示。

代码 4-1 生成的指令

```

1  [
2    {
3      "command" : "change",
4      "paras" : [ ["Background", "Upper-clothes"] , "blue\u2022sky,\u2022white\u2022T-shirt"]
5    }
6  ]

```

#### 4.1.2.2 JSON 指令校验

由于大语言模型生成指令不稳定，需要对生成指令的合规性进行校验。校验规则存储为一个 JSON 文件，以修改非面部和面部的指令为例，校验规则如代码 4-2 所示：

代码 4-2 指令校验规则

```

1  {
2    "face": {
3      "paras_type": [
4        "<class\u2022'str'\u2022>"
5      ],
6      "paras_enum": null,
7      "paras_min": null,
8      "paras_max": null,
9      "combine": true,
10     "priority": 1
11   },
12   "change": {
13     "paras_type": [
14       "<class\u2022'list'\u2022>",
15       "<class\u2022'str'\u2022>"
16     ],
17     "paras_enum": null,
18     "paras_min": null,
19     "paras_max": null,
20     "combine": true,
21     "priority": 1
22   }
23 }

```

指令校验的算法如算法 6 所示。

---

**算法 6 JSON 指令校验算法**

---

输入：待校验的指令 *JsonCommands*、校验规则 *Rules*

输出：校验后的指令 *ValidJsonCommands*

```

1: for Command in JsonCommands do
2:   if Command satisfy Rules then
3:     Add Command to ValidJsonCommands
4:   end if
5: end for

```

---

**4.1.2.3 多指令处理**

由于一次执行可能会涉及到多个指令，会遇到指令重复、指令优先性等问题，所以会对需要执行的指令进行合并与排序。指令合并的算法如下：

---

**算法 7 多指令处理算法**

---

输入：待处理的指令 *OriginCommands*、指令合并规则 *Rules*、指令优先性 *Priority*

输出：处理后的指令 *ProcessedCommands*

```

1: for Command in OriginCommands do
2:   if Same Command type already in ProcessedCommands then
3:     Combine Command to the same one in ValidJsonCommands use Rules
4:   end if
5: end for
6: 根据 Priority 对 ValidJsonCommands 进行排序

```

---

**4.1.2.4 图像模型请求参数生成**

图像模型请求参数生成较为复杂，对于某个参数，其可能来源于 GUI 中可修改的设置，可能来源于指令，可能来源于模版，否则设置为默认参数。由于每个参数来说，其来源的优先性可能不一致，因此设计了算法 8 来生成图像模型请求参数。

**4.1.3 图像修改建议**

本项目提供了根据图像自动生成图像建议的功能。由于传统的大语言模型只能接受文本输入，因此本项目采用了 GPT4V 来自动生成图像修改建议。

GPT-4V 是由 OpenAI 开发的多模态大型语言模型，是 GPT 系列基础模型的第四代。该模型具有视觉能力，可以将图片作为输入，进行各种任务，例如描述图片中的幽默、总结截图文本、回答包含图表的考试题目等。

用户通过 GUI 界面的 Advise 按键，可以生成建议并将其转换为指令。

---

**算法 8** 图像模型请求参数生成算法

---

输入：指令 *Command*、设置 *Settings*、默认参数 *Default*、参数来源优先性 *PriorityRules*

输出：图像模型请求参数 *Parameters*

```

1: for ParaKey in Parameters do
2:   Get Template from Command
3:   if ParaKey found in Command or Settings or Template then
4:     Choose the highest priority source use PriorityRules[ParaKey]
5:   else
6:     Set this parameter to Default[ParaKey]
7:   end if
8: end for

```

---

## 4.2 Middleware 的构建

在本项目中，Middleware 作为核心组件，通过整合来自不同平台的 API，为 GUI 提供了统一且易于接入的 API 服务。其整合了多个图像生成和大语言语言模型，通过统一的接口设计，使得 GUI 能够方便地调用所需的功能。此外，Middleware 采用了 Golang 语言和 Beego 框架，不仅保证了 API 服务的高并发处理能力和稳定性，还通过模块化的设计提高了系统的可维护性和可扩展性。主要 API 服务包括 Stable Diffusion 和 DALL-E2 模型、图像分割模型，以及多种大语言模型。这样的架构设计不仅优化了开发效率，也确保了系统的稳定运行和长期发展。

### 4.2.1 对多个平台的 API 进行配置和整合

Middleware 通过整合不同平台如图像生成模型、语言模型等的 API，使得 GUI 开发者可以通过一个统一的接口调用多种功能。这种整合不仅包括 API 的聚合，还涉及统一 API 调用的风格和路由规范，不仅保证了 API 服务的高稳定性和可靠性，还便于日后的维护和扩展。例如，无论是调用 ChatGLM2-6B 模型还是多种 GPT 模型，GUI 都能通过相同的结构化请求方式访问不同的服务。

### 4.2.2 使用 Beego 框架提供 API 服务

本项目使用 beego 框架构建了 Middleware 组件提供了多个 API 服务，以便 GUI 可以高效地使用各种模型和工具。*PostSDTxt2Img* 和 *PostSDImg2Img* 是通过 Stable Diffusion 模型来生成或修改图像的 API，这使得用户能够通过简单的 API 调用，进行复杂的图像生成和编辑操作。*GetLoras* 这个 API 用于获取 Stable Diffusion 模型可用的 LoRA 模型列表。*PostDALLE2Edit* 利用 DALL-E2 模型修改图片，这进一步扩展了图像处理的能力，在 Stable Diffusion 模型不可用时作为替代。*PostHuggingFaceImgSegment*

API 通过可在部署在本地的分割模型不可用时通过调用 Hugging Face 上的图像分割模型 API 来实现图像分割。在文本处理方面, *PostGPT3Dot5Turbo*、*PostGPT4Turbo* 和 *PostGPT4V* 等 API 利用 OpenAI 的不同版本 GPT 模型来处理指令理解和生成任务, *PostChatGLM2\_6B* 可调用微调后的 ChatGLM2-6B 模型来进行指令生成。

## 4.3 LLM 的微调

大语言模型 (LLM) 的微调是一种在预训练模型基础上通过特定数据集进一步训练的过程, 用于优化模型在特定任务或场景中的表现。特别是对于参数量较小的 LLM, 微调不仅可以提升其性能, 还能增强其针对具体任务的适应性和泛化能力。微调的一个主要优势是性能提升, 即使是较小的模型, 通过针对性的微调, 可以在特定任务上实现甚至超过大型模型未经微调时的性能, 可见微调能够根据具体需求调整模型的行为, 使其更加专注于特定的输出目标。微调技术提供了一种有效途径, 通过少量的定制化数据提升模型的应用性能, 特别是在参数量较小的模型中, 这种优势尤为显著。

原始的 ChatGLM2-6B 模型已经通过大规模数据预训练, 具备了强大的语言理解和生成能力, 但受制于参数量较小, 其在本项目需要高精准度的指令生成任务上表现不佳, 因此本项目通过自动生成并进行筛选的高质量指令生成微调数据集对 ChatGLM2-6B 进行微调以进一步提升模型在指令生成任务的表现。

本项目利用开源项目 LLaMA-Factory<sup>1</sup>和本项目中自动生成的指令生成任务数据集对 ChatGLM2-6B 模型进行微调。

### 4.3.1 LLM 微调数据集生成与性能评估方法

#### 4.3.1.1 微调数据集生成

在大语言模型 (LLM) 的微调过程中, 数据集的功能和作用至关重要。微调数据集不仅提供了模型训练所需的具体数据, 还直接影响了模型微调后的性能和适应性。对于参数量较小的模型, 高质量的微调数据集尤其重要, 因为这些模型通常缺乏足够的参数量来从大规模数据中学习复杂的特征。通过高质量的微调数据集, 参数量较小的大语言模型可以有效地提高这些模型的学习效率和最终性能。构造高质量的微调数据集涉及两个关键步骤: 数据收集和数据预处理。数据收集需要确保获得足够多的、具有代表性的数据, 这些数据能够覆盖模型在实际应用中可能遇到的各种情况。数据预处理是将收集到的原始数据转换成模型可以直接处理的格式, 这可能包括数据清洗、特征提取、标签编码等。

由于没有适用于本任务的开源数据集, 本项目尝试建立一个自动化工作流程, 通过利用 ATR Dataset<sup>2</sup>调用多个模型和一定的校验规则来生成所需的数据并整合为一个数据

<sup>1</sup><https://github.com/hiyouga/LLaMA-Factory>

<sup>2</sup><https://github.com/lemondan/HumanParsing-Dataset>

集。数据集生成的流程如图 4-6。同时，在系统使用过程中产生的数据可设置是否保存，这些数据也会在运行数据集生成脚本添加到数据集中。

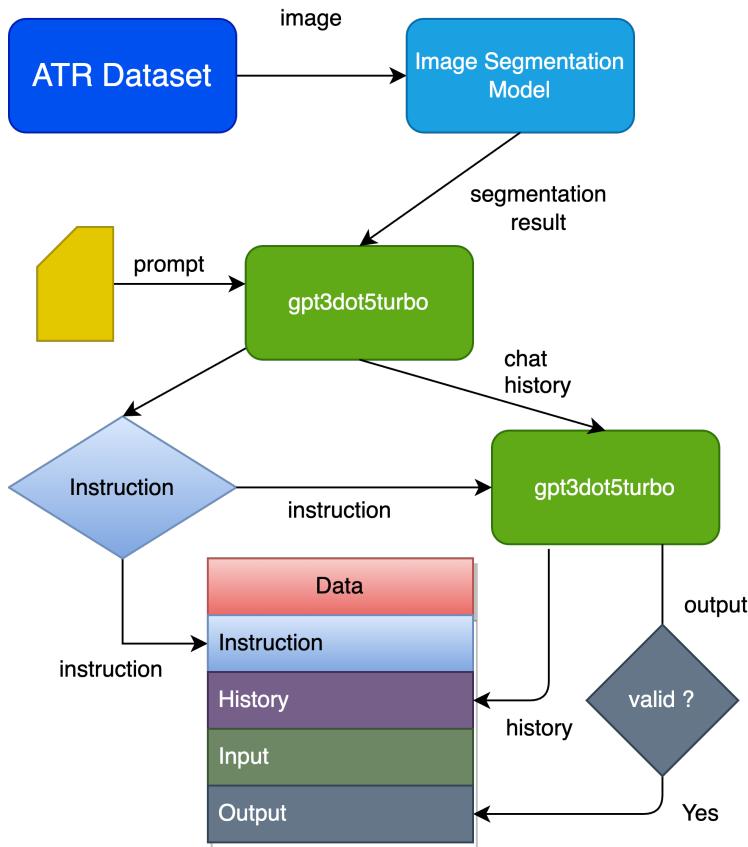


图 4-6

通过建立的自动化工作流程，本项目创建了一个包含 3000 个样本的数据集，数据集中的每一项数据都经过合法性校验以保证数据集的质量。数据集的格式满足 LLaMA-Factory 的要求并可使用此数据集对 ChatGLM2-6B 进行微调。

#### 4.3.1.2 LLM 指令生成任务性能评估方法

性能评估是大语言模型（LLM）开发和应用的关键环节，尤其是在模型的实用性和可靠性方面。有效的评估方法可以帮助研究者和开发者了解模型在特定任务上的表现，从而进行进一步的优化和调整。

由于需要一种直观的性能评估方法来对不同的大语言模型在指令生成任务上的性能进行评估，本项目采用生成指令的合法率对不同的大语言模型在指令生成任务上的性能进行评估。

#### 4.3.2 ChatGLM2-6B 针对指令生成任务的微调

LoRA: Low-Rank Adaptation of Large Language Models<sup>[40]</sup> 这篇论文提出了一种新颖且高效的 LoRA 微调方法，用于微调大型预训练语言模型以适应特定任务。传统的微调

方法往往需要重新训练模型的所有参数，而全参数训练的方法在模型参数规模庞大时需要巨大的量。算法主要通过在 Transformer 模型的每一层中注入低秩的分解矩阵来更新权重，而不改动预训练的权重，其具体算法描述见算法 9。

---

#### 算法 9 LoRA: 大型语言模型的低秩适配

---

**输入:** 预训练权重矩阵  $W_0 \in \mathbb{R}^{d \times d}$ , 输入  $x$ , 秩  $r$

**输出:** 调整后的输出  $h$

- 1: 初始化低秩矩阵  $A \in \mathbb{R}^{r \times d}$  和  $B \in \mathbb{R}^{d \times r}$
  - 2: 冻结预训练权重  $W_0$
  - 3: **for** 对于每个训练步骤: **do**
  - 4:   计算更新矩阵  $\Delta W = BA$
  - 5:   应用更新:  $W = W_0 + \Delta W$
  - 6:   计算输出:  $h = Wx$
  - 7: **end for**
  - 8: 优化参数  $A$  和  $B$  以最小化损失函数
- 

本项目使用生成的指令生成任务数据集结合 LoRA 方法，通过开源项目 LLaMA-Factory<sup>1</sup>对 ChatGLM2-6B 模型进行微调。LoRA 微调在大语言模型上的训练损失随训练步数变化的情况如图 4-7 所示。从图中可以看出，最初损失值很高，但随着训练步数的增加，损失值迅速下降，特别是在前 50 步之内下降最为显著。在经过约 50 步之后，损失下降的速度开始放缓，但仍然持续下降，表明模型继续从训练数据中学习。在约 200 步之后，损失曲线趋于平缓，说明模型已经接近收敛，额外的训练步骤在减少损失方面的效果变得有限。图中还展示了一个平滑处理的损失曲线，更清晰地显示了训练过程的整体趋势，而不是每一步的波动。

#### 4.3.3 各个 LLM 在本任务下的性能评估

结合本项目的 LLM 指令生成任务性能评估方法，本项目对不同的大语言模型进行了指令生成任务性能评估。各个模型的性能表现如表 4-2 和图 4-8 所示。

可观察到原始的 ChatGLM2-6B 模型在未经过微调时，在指令生成任务中的合格率仅为 6%。这一低下的性能表现暴露了 ChatGLM2-6B 模型在没有针对性训练的情况下难以完成需要高精准度指令生成任务。通过使用 LoRA 方法进行微调，模型性能随着微调步骤的增加显著提升。当微调步骤增至 80 步时，合格率提升至 77.8%；而在经过 200 步的微调后，合格率达到 95.8%，并在 300 步训练后稳定在 96% 左右。与 ChatGLM2-6B 相比，GPT3.5Turbo 和 GPT4Turbo 在无需额外训练的情况下即可达到分别为 97% 和 99% 的高合格率。

---

<sup>1</sup><https://github.com/hiyouga/LLaMA-Factory>

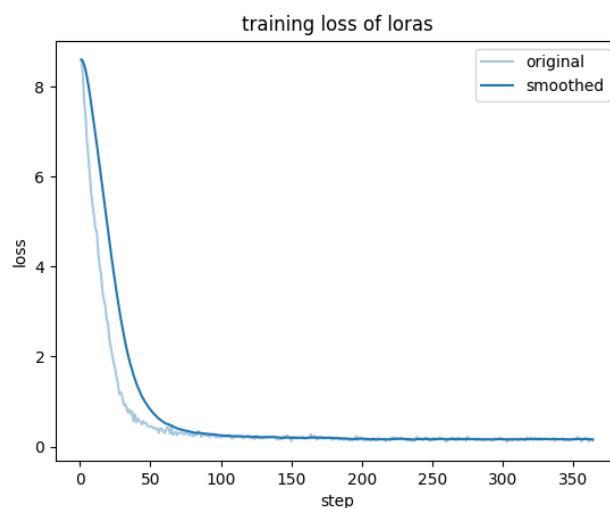


图 4-7 step-loss 图

表 4-2 LLM 指令生成性能

模型	测试样本数	合格率
GPT3.5Turbo	100	97%
GPT4Turbo	100	99%
ChatGLM2-6B(Origin)	100	6%
ChatGLM2-6B(LoRA trained 50 steps)	1000	19.1%
ChatGLM2-6B(LoRA trained 60 steps)	1000	43.7%
ChatGLM2-6B(LoRA trained 80 steps)	1000	77.8%
ChatGLM2-6B(LoRA trained 100 steps)	1000	88.1%
ChatGLM2-6B(LoRA trained 150 steps)	1000	93.9%
ChatGLM2-6B(LoRA trained 200 steps)	1000	95.8%
ChatGLM2-6B(LoRA trained 250 steps)	1000	95.2%
ChatGLM2-6B(LoRA trained 300 steps)	1000	96.7%

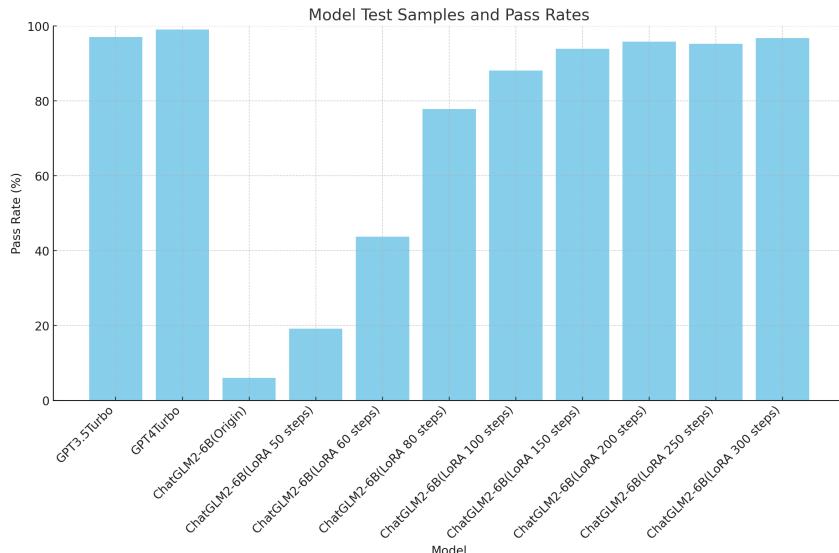


图 4-8 LLM 指令生成性能

#### 4.4 Stable Diffusion 及扩展的使用

Stable Diffusion 是一种基于深度学习的图像生成模型，能够根据文本描述生成高质量的图像。该模型利用条件生成技术，允许用户通过简单的文本指令来引导图像的生成过程。这种技术在艺术创作、媒体娱乐、广告以及数字营销等多个领域显示了广泛的应用潜力。

除了 Stable Diffusion 的基础功能外，本项目还集成了多个扩展，如 ControlNet 和 roop。ControlNet 允许用户对生成的图像进行更精确的控制，通过这种方式 ControlNet 增强了图像的细节质量和一致性，使生成的图像更贴近用户的具体需求。roop 专注于提供面部替换功能，这一功能特别适用于需要在图像中修改人物面部特征的场景。roop 通过简单的操作接口，使得用户能够轻松地将一张人脸图像替换到另一张脸上，而不需要复杂的图像处理知识背景。

sd-webui-controlnet<sup>1</sup> 使用了 ControlNet 的原理，旨在增强原有 Stable Diffusion 模型的图像生成控制能力。通过集成一个额外的控制网络（ControlNet），允许用户精确指导图像的具体内容，显著提升了生成图像的细节质量和一致性。

sd-webui-roop<sup>2</sup> 基于 DeepFake<sup>[41]</sup>，允许用户在图片中进行面部替换，简化了面部交换的过程，无需训练特定的模型，大大降低了使用复杂度。

系统还提供了丰富的参数调整选项，如调整噪声去除强度。这些功能为用户提供了广泛的自定义空间，可以根据具体的应用场景调整生成效果，以满足不同的视觉表达需求。Stable Diffusion 模型及其扩展的结合使用，为本项目的图像编辑方面提供了强大的功能和更高的灵活性。

<sup>1</sup><https://github.com/Mikubill/sd-webui-controlnet>

<sup>2</sup><https://github.com/s0md3v/sd-webui-roop>

#### 4.4.1 Stable Diffusion API 的使用

本项目通过将开源项目 stable-diffusion-webui<sup>1</sup>部署在揽睿星舟机器学习平台<sup>2</sup>，通过 API 来调用 Stable Diffusion 模型，主要使用的参数见表 4-3。

表 4-3 主要使用的 Stable Diffusion webui img2img API 参数

参数	描述	形式
prompt	输出图像的期望修改或主题	str
negative_prompt	生成图像中应避免的内容	str(base64 Image)
mask	选择性编辑或生成的区域的图像遮罩	str
inpainting_fill	修补时的填充方法	int
inpaint_full_res	是否在图像的全分辨率下应用修补	bool
inpaint_full_res_padding	使用全分辨率时修补区域周围的填充	int
inpainting_mask_invert	是否反转修补遮罩	bool
mask_blur	遮罩边缘的模糊量	float
denoising_strength	去噪强度	float
sampler_index	采样算法	str
seed	初始化随机数生成器的值	int
steps	生成图像时的步骤数	int
width	输出图像的宽度	int
height	输出图像的高度	float
cfg_scale	输入提示的权重	float
restore_faces	是否修复生成图像中的面部	bool
alwayson_scripts	插件参数	dict

#### 4.4.2 ControlNet 的使用及效果

sd-webui-controlnet<sup>3</sup>是一个用于 stable-diffusion-webui 的扩展，允许用户通过添加额外的条件来控制扩散模型的行为，从而增强生成图像的精确度和控制性。这一扩展可以实时添加到原始的 Stable Diffusion 模型中，不需要进行合并处理。本项目使用该插件以保持原始图像的主要轮廓和布局不受改变，其实现的效果如图 4-9。

<sup>1</sup><https://github.com/AUTOMATIC1111/stable-diffusion-webui>

<sup>2</sup><https://www.lanrui-ai.com>

<sup>3</sup><https://github.com/Mikubill/sd-webui-controlnet>



图 4-9 ControNet 效果: (a)原始图像, (b)未使用 ControlNet, (c)使用 ControlNet

#### 4.4.3 Roop 的使用及效果

roop<sup>1</sup>是一个用于 stable-diffusion-webui 的扩展，提供面部替换的功能。这个扩展需要一张原始图像和目标图像，并能将原始图像的面部替换到目标图像上。其实现的效果如图 4-10。



图 4-10 roop 效果: (a)原始图像, (b)目标图像, (c)结果

## 4.5 本章小结

本章中详细介绍了基于大语言模型（LLM）的交互式多模态图像编辑系统的设计与实现。本章首先阐述了图形用户界面（GUI）的构建过程，接着描述了中间件的开发过程，还探讨了 LLM 的微调技术及其在本系统中的应用，以及如何利用 Stable Diffusion 模型和其他技术来增强图像编辑的功能。这些元素共同构成了一个高效、直观的图像编辑系统。

<sup>1</sup><https://github.com/s0md3v/sd-webui-roop>

## 第五章 系统实现效果与使用

系统实现的核心在于其高度集成的图形用户界面 (GUI)，该界面不仅简洁易用，还充分支持复杂的图像处理操作。GUI 设计考虑了用户的操作习惯，提供了直观的图像上传、编辑和预览功能。用户可以通过简单的点击和拖拽操作，实现图像的上传和编辑。GUI 还集成了多个图像生成模型和大语言模型并提供控制选项，如模型选择、参数调整等，使用户能够根据具体需求调整图像处理流程。

### 5.1 系统实现效果

系统实现了一个简单易用的 GUI，可通过特定端口进行访问。GUI 的总体效果如图 5-1 所示。

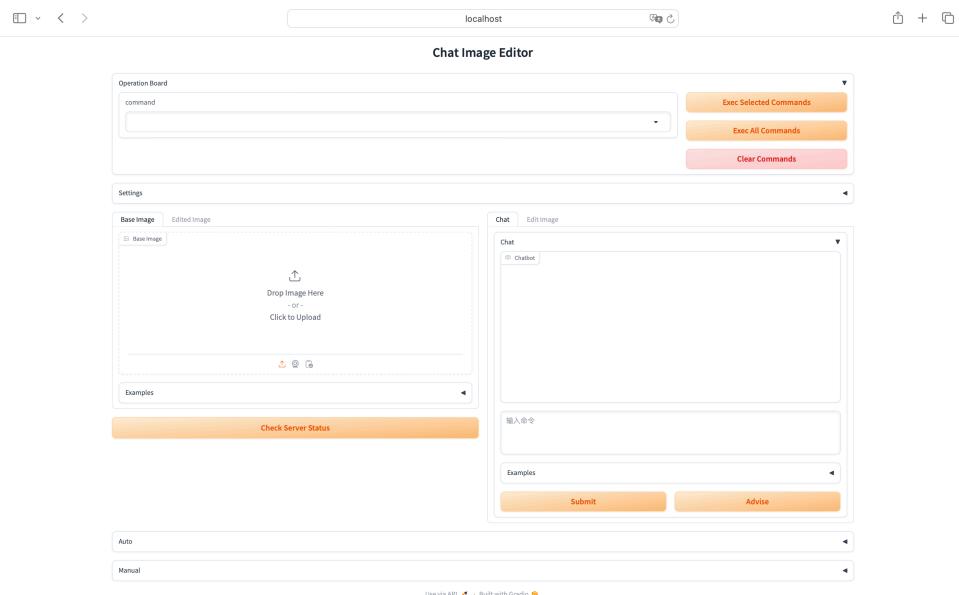


图 5-1 GUI 总体效果

可使用 GUI 的 Auto 模块进行 LLM 微调数据生成和 LLM 指令生成任务性能测试，其实现效果如图 5-2 所示。

可使用 GUI 的 Manual 模块查看 GUI 使用方法，其实现效果如图 5-3 所示。

使用 GUI 进行基于 LLM 的交互式图像编辑时，系统主要模块效果如图 5-4 所示。

### 5.2 系统使用方法

#### 5.2.1 系统部署

本项目提供了 Docker、Kubernetes 等多种常用的项目部署方式以适应不同的用户需求和操作环境。如果选择通过 Docker 进行部署，用户可以根据需求选择不同的镜像。通过运行命令 `docker run --name multimodal -p 27777:80 binciluo/multimodal:latest`，可以在



图 5-2 Auto 模块

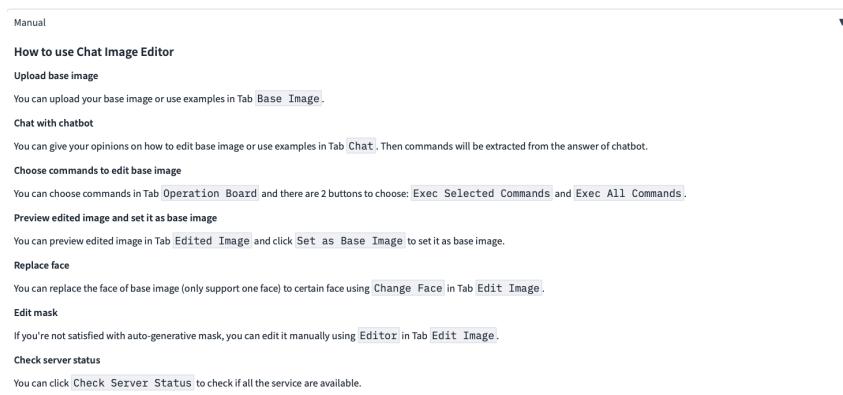


图 5-3 Manual 模块

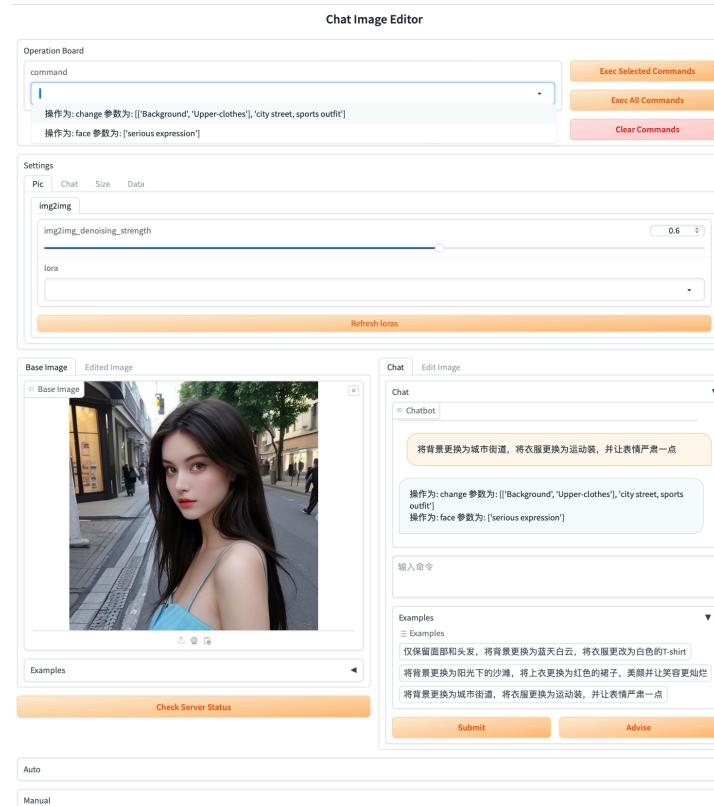


图 5-4 交互式图像编辑

本地部署图像分割模型。如果希望通过 Hugging Face API 使用图像分割功能以在性能受限的设备上运行，则可以使用命令 `docker run -name multimodal -p 27777:80 binciluo/multimodal:mini_latest`。部署完成后，用户可通过访问本地地址 `127.0.0.1:27777` 来使用服务。对于 Kubernetes 部署，用户需要先切换到包含 Kubernetes 配置文件的目录，使用 `kubectl apply -f pod.yaml` 命令部署服务（对于使用 Arm 架构的用户，则需使用 `kubectl apply -f pod_arm.yaml`）。非 Linux 用户需运行 `kubectl port-forward mm-service-pod -n default 27777:27777` 以访问服务。服务可通过 `127.0.0.1:27777` 地址访问。

项目还支持本地直接运行。用户需要先安装 Golang 和 Beego 框架，然后安装 Python 所需的依赖包。首先安装 Golang，接着通过命令 `go install github.com/beego/bee/v2@latest` 安装 Beego，然后运行 `pip install -r gradio_web/requirements.txt` 安装 Python 依赖。最终，通过运行脚本 `runner.sh` 启动服务，或者设置环境变量 `SEG_MODEL_ENV='local'` 并运行 `runner_local.sh` 脚本以在本地使用图像分割模型。

### 5.2.2 GUI 使用说明

1. 上传基础图像：您可以上传您的基础图像，或者使用在 Base Image 中的 Examples 提供的示例图像。图像上传后，系统会自动地进行图像分割任务并在 Chat 界面显示分割结果。
2. 与 LLM 对话您可以就如何编辑基础图像提出您的看法，或者使用在 Chat 中的 Examples 提供的示例。LLM 的回复会被识别是否含有指令，若有指令存在则会对指令进行提取并自动添加至 Operation Board 中。
3. 选择指令来修改基础图像您可以在 Operation Board 中选择单个或多个指令，并有两个按钮可供选择：Exec Selected Commands 和 Exec All Commands。Exec Selected Commands 会执行选中的指令，而 Exec All Commands 会执行所有指令。系统会对这些指令进行合并与排序等预处理，然后生成对应的参数向图像编辑模型发起请求。
4. 预览编辑后的图像并设置为基础图像您可以在 Edited Image 标签中预览编辑后的图像，并点击 Set as Base Image 将其设置为新的基础图像。
5. 替换面部您可以使用 Edit Image 标签中的 Change Face 功能，将基础图像的面部（仅支持单一面部）替换为特定面部。
6. 编辑遮罩如果您对自动生成的遮罩不满意，可以在 Edit Image 标签的 Editor 中手动编辑，然后重新运行之前的指令。
7. 检查服务器状态您可以点击 Check Server Status 来确认所有服务是否可用。系统会在大约三秒后弹出提示框显示哪些服务目前不可用。
8. 自动生成 LLM 微调数据您可在 Auto 模块中的 Gen Data 选择图片，输入 OpenAI 的 API Key 并指定生成数量和线程数以生成 LLM 微调数据。
9. 对 LLM 指令生成合格率进行测试您可在 Auto 模块中的 Test LLM 选择测试集并指定测试样本数量和线程数以对 LLM 指令生成合格率进行测试。

### 5.3 本章小结

在本章中，我们展示了基于 LLM 的交互式多模态图像编辑系统的实际运行效果和用户使用方法，提供了详细的用户操作指南，确保用户能够充分利用系统功能进行图像编辑。

## 第六章 项目管理与维护

### 6.1 代码管理

Git 是一个开源的分布式版本控制系统，其允许多个开发者在共同的代码基础上工作，同时能够追踪和记录所有文件的历史变更，兼具高性能与灵活性。Git 能处理从小到大的项目，让开发者能够在本地机器上工作，并保持代码的多个版本，以便在不同的分支上进行试验和开发新功能而不影响主代码库。

GitHub<sup>1</sup>是一个基于 Git 的代码托管和协作平台，其为开发者提供了一个强大而便捷的环境来管理代码和协作，不仅能够追踪和记录代码的变更历史以确保代码的完整性和回溯能力，还能通过分支管理支持多线程的工作流以允许多个开发者同时推进不同的功能。GitHub 的 pull 请求机制促进了团队成员之间的代码审查和讨论，不仅提高了代码质量也加强了团队协作能力。GitHub 的集成系统支持持续集成和持续部署流程，与各种开发工具链的无缝连接，使得项目管理更加高效。由于其承载了众多开源项目，GitHub 成为开发者的一个展示和交流的平台，促进了知识共享和技术交流。

为了便于进行代码管理和版本控制，本项目在 GitHub 上创建了一个仓库<sup>2</sup>，结合 GitHub 的其他功能，将其发展成了功能完整、文档详细的开源项目。

### 6.2 自动化测试

自动化测试是一种利用软件来控制执行测试的过程，其能自动比较实际的运行结果与预期结果来验证被测软件功能是否符合预期。自动化测试的主要功能是提高测试的效率和覆盖率，其可以快速地执行大量的测试用例，并且可以反复运行这些测试，在软件开发的早期发现缺陷，从而减少修复缺陷的成本并确保软件在开发迭代中未引入错误。自动化测试还可以将测试人员从繁琐的手动测试工作中解脱出来，使他们有更多时间专注于更复杂的测试任务。在持续集成和持续部署（CI/CD）的实践中，自动化测试是不可或缺的一环，它提高了软件交付的速度和质量，是现代软件开发流程中的关键组成部分。

GitHub Actions 是 GitHub 提供的一个持续集成与持续部署（CI/CD）平台，允许用户在代码仓库中直接自动化、自定义地执行特定的软件开发工作流程相关任务。用户可定义一系列的事件和操作，当如代码推送、合并请求等指定事件发生时，GitHub Actions 会自动运行对应的构建代码、运行测试、部署到生产环境等任务。GitHub Actions 使得开发者无需离开 GitHub 环境就能自动化处理软件的构建、测试和部署过程，从而大幅提升开发的效率。它支持多种操作系统，提供了大量现成的 Actions 供用户使用，并且允许创建私有的、自定义的 Actions。作为 CI/CD 的解决方案，GitHub Actions 简化了开

<sup>1</sup><https://github.com>

<sup>2</sup><https://github.com/BinciLuo/multimodal-service>

发流程，加快了从编写代码到部署产品的过程，同时还提高了软件的质量和交付的可靠性。

本项目使用 GitHub Actions 对代码中的部分模块进行自动化测试以保证代码的正确性和项目的稳定性。当有新的 pull 请求对 main 或 dev 分支进行更新时，自动化测试工作将在最新版本的 Ubuntu 运行环境上执行。其首先会使用 actions/checkout@v3 获取最新的仓库代码，利用 actions/setup-python@v3 来设置 Python 3.10 版本的 Python 环境。当设置好 Python 环境后，需要安装测试所需的依赖。在 gradio\_web 目录下首先升级 pip，然后安装本项目自动化测试所需的代码检查和测试框架 flake8 和 pytest，如果存在 requirements.txt 文件，还会安装该文件中列出的依赖。最后，流程将继续在 gradio\_web 目录下执行名为 test\_utils.py 的测试脚本。

### 6.3 持续集成与持续部署

持续集成 (Continuous Integration, CI) 和持续部署 (Continuous Deployment, CD) 是现代软件开发流程中的关键部分。CI 的核心是自动化地将代码变合并到特定分支时自动运行构建和测试流程，这样可以迅速发现并解决集成错误，提高代码质量，缩短反馈周期。CD 扩展了 CI 的概念，不仅包括自动化测试，还包括自动化部署过程，确保经过测试的代码可以被自动地部署到指定环境中。这使得产品能够快速迭代，缩短从开发到产品投放市场的时间，同时减少了部署过程中的人为错误，提升了软件交付的速度和安全性。CI/CD 通过自动化的流程减少了手动工作以使开发团队更加专注于功能开发和创新。

本项目使用 GitHub Actions 进行 CI 和 CD 流程。除了测试外，本项目还有几个关键的 CI/CD 流程，主要包括部署将项目部署到 Azure Web 应用和 Docker 镜像构建。

名为“Build and deploy container app to Azure Web App - gradio-app”的 Action 在代码推送到 main 分支时手动触发，自动化了在 Azure Web App 上的部署过程。其首先构建 gradio\_web 的 Docker 镜像，并将其推送到 DockerHub，随后将镜像部署到 Azure 的生产环境，从而实现高效和一致的应用发布。名为“Build and deploy container app to Azure Web App - middleware-app”的 Action 以相近的方式实现了 Middleware 在 Azure Web App 上的自动化部署。

名为“Docker Image CI”的 Action 主要用于构建和推送 Docker 镜像到 DockerHub。当代码被推送到 main 分支时，此工作流程触发并执行以下操作：使用最新的 Ubuntu 环境，首先通过 GitHub Secrets 进行 Docker 登录，然后分别从 docker/Dockerfile 和 docker/DockerfileMini 两个文件构建构建标准镜像和更小的 Mini 镜像，其区别为标准镜像使用本地的模型进行图像分割而 Mini 模型使用 HuggingFace<sup>1</sup>的 API 进行图像分割。这些镜像将在构建完成后被标记并推送到 DockerHub 账户下，确保最新的容器镜像版本可供部署和分发。此自动化流程加快了软件的交付速度，保证了镜像的最新状态和可用性。

<sup>1</sup><https://huggingface.co>

名为“Docker Image CI for ARM64”的 Action 通过相同的方式实现了适用于 arm64 架构的镜像的构建与发布。

## 6.4 本章小结

在本章中，我们探讨了项目管理与维护的关键策略和实践，确保了基于 LLM 的交互式多模态图像编辑系统的高效开发和长期可持续性发展。本章详细介绍了项目的代码管理方式、自动化测试以及持续集成与部署的实施方法。这些管理和维护策略不仅提高了开发过程的效率，也确保了系统在实际部署后的稳定性和可靠性，为未来可能的功能扩展和技术升级打下了坚实的基础。

## 第七章 总结及未来展望

### 7.1 总结

本系统的研发始于当前图像编辑工具的局限性，这些工具往往需要用户具备专业知识和技能，且操作复杂，难以满足非专业用户的需求。为了解决这些问题，本项目通过打通最新的大语言模型和图像生成模型，开发了一个既强大又用户友好的图像编辑系统。系统的核心在于其能够通过简单的与大语言模型聊天的方式来自动化地生成图像编辑的指令从而调用图像生成模型来执行复杂的图像编辑任务。从系统架构层面，打通大语言模型和图像生成模型需要两个主要组件的协同工作：一个直观的图形用户界面（GUI）和一个功能强大的中间件（Middleware）。

图形用户界面（GUI）的设计充分考虑了易用性和高效性，使得即使是没有图像编辑经验的用户也能够轻松使用。通过各个简洁明了的模块，用户可以执行包括文本交互、图像自动遮罩、更换面部、参数设置等多种任务。GUI 既为用户提供了足够简单易用的自动化操作，也能让用户对如遮罩生成等高精度要求的操作进行手动的调整。GUI 不仅提高了操作的直观性，还通过使用中间件（middleware）访问一些对资源需求较大的模型，极大地降低了对用户设备性能的需求。

项目的中间件（Middleware）部分是整个系统的枢纽。它整合了多个不同平台、不同请求格式、不同返回格式的 API，包括图像生成模型如 Stable Diffusion、DALL-E2 和大语言模型如 ChatGLM2-6B、GPT-3.5 Turbo 和 GPT-4 Turbo。中间件（Middleware）的设计保证了这些模型的高效协同工作，支持了 GUI 所需的从文本交互到图像编辑的一系列高级功能。此外，中间件（Middleware）还处理所有后端逻辑，包括一对多且互相隔离的服务、以及用户请求的响应。

系统创新地引入了 LoRA 进行大语言模型的微调，通过训练专门的 LoRA 模型，原本在本任务下表现较差的 ChatGLM2-6B 模型能够理解复杂的用户输入并精准地将其转化为指令，然后自动对指令进行抽取并执行高质量的图像编辑任务。微调数据集的自动生成是其中的一个重要环节，系统采用了结合多个模型输出和验证规则的自动化工作流程。在已有的图像数据集下，利用现有的图像分割模型和文本生成模型，自动地生成图像编辑的要求、指令，并进行数据合格校验，生成了大量高质量的训练数据。系统设计了自动化的测试流程对各个大语言模型的指令生成能力进行评估，并对 ChatGLM2-6B 的多个微调模型和 GPT3.5 turbo、GPT4 Turbo 进行评估，以便对不同的模型的指令生成能力进行量化。从结果中可观察到 GPT3.5 turbo、GPT4 Turbo 在不进行微调的情况下就已经具有很强的能力，而参数量较小、初始能力极弱的 ChatGLM2-6B 模型在微调后在指令生成上也能达到接近 GPT3.5 turbo 的能力。

## 7.2 未来展望

随着最近几年 GPU 和 TPU 等专用硬件的发展，深度学习的训练和推理速度得到了极大的提升，深度学习在图像和语音识别、自然语言处理和自动驾驶等许多领域都取得了显著进展。在文本生成和图像生成任务上，深度学习技术已从理论探索逐步过渡到实际应用，图像生成技术已经实现了从简单的图像生成到复杂的场景重构的跨越，在这一领域，生成对抗网络（GANs）和最新的扩散模型都已经能够生成高质量的图像内容。而 OpenAI 的 GPT 系列和清华大学的 ChatGLM2-6B 等大语言模型也已经能够理解并生成复杂的自然语言文本。在这些新技术的加持下，通过一定方法对大语言模型和图像生成模型进行链接，从而使基于 LLM 的交互式图像编辑系统成为可能。本项目尝试了基于特定指令链接大语言模型和图像生成模型的方法，并搭建了基础框架且得到了优异的效果。

大语言模型和图像生成技术的结合带来了许多机遇，但也存在不少技术和伦理方面的挑战。如何确保生成的内容的准确性和适当性，防止生成有偏见或不当信息的风险，数据隐私和安全成为重点关注的问题，必须妥善处理，确保内容合规性和用户信息受到保护的同时不妨碍技术的有效应用。

随着深度学习技术的继续发展，未来的交互式技术必将更加智能和高效。大语言模型和图像生成技术的结合不仅能提升用户在图像编辑领域的体验，还将开启全新的应用，为创新和改进现有服务提供强大动力。

## 参考文献

- [1] Zhan Fangneng, Yu Yingchen, Wu Rongliang et al. Multimodal image synthesis and editing: A survey [J]. arXiv preprint arXiv:2112.13592. 2022.
- [2] Esser Patrick, Rombach Robin, Blattmann Andreas et al. Imagebart: Bidirectional context with multinomial diffusion for autoregressive image synthesis [J]. Advances in Neural Information Processing Systems. 34. 2021: 3518–3532.
- [3] Lewis Mike, Liu Yinhuan, Goyal Naman et al. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension [J]. arXiv preprint arXiv:1910.13461. 2019.
- [4] Ling Huan, Kreis Karsten, Li Daiqing et al. Editgan: High-precision semantic image editing [J]. Advances in Neural Information Processing Systems. 34. 2021: 16331–16345.
- [5] Mansimov Elman, Parisotto Emilio, Ba Jimmy Lei et al. Generating images from captions with attention [J]. arXiv preprint arXiv:1511.02793. 2015.
- [6] Barrett William A, Cheney Alan S. Object-based image editing [C]. In 29th Annual Conference on Computer Graphics and Interactive Techniques. 2002 : 777–784.
- [7] Portenier Tiziano, Hu Qiyang, Szabo Attila et al. Faceshop: Deep sketch-based face image editing [J]. arXiv preprint arXiv:1804.08972. 2018.
- [8] Oh Byong Mok, Chen Max, Dorsey Julie et al. Image-based modeling and photo editing [C]. In 28th Annual Conference on Computer Graphics and Interactive Techniques. 2001 : 433–442.
- [9] Perarnau Guim, Van De Weijer Joost, Raducanu Bogdan et al. Invertible conditional gans for image editing [J]. arXiv preprint arXiv:1611.06355. 2016.
- [10] Zhu Jiapeng, Shen Yujun, Zhao Deli et al. In-domain gan inversion for real image editing [C]. In European Conference on Computer Vision. 2020 : 592–608.
- [11] Di Martino J Matías, Facciolo Gabriele, Meinhardt-Llopis Enric. Poisson image editing [J]. Image Processing On Line. 6. 2016: 300–325.
- [12] Elder James H, Goldberg Richard M. Image editing in the contour domain [C]. In IEEE Computer Society Conference on Computer Vision and Pattern Recognition. 1998 : 374–381.
- [13] Farbman Zeev, Fattal Raanan, Lischinski Dani. Diffusion maps for edge-aware image editing [J]. ACM Transactions on Graphics (TOG). 29 (6). 2010: 1–10.
- [14] Ding Ning, Qin Yujia, Yang Guang et al. Parameter-efficient fine-tuning of large-scale pre-trained language models [J]. Nature Machine Intelligence. 5 (3). 2023: 220–235.
- [15] Radiya-Dixit Evani, Wang Xin. How fine can fine-tuning be? learning efficient language models [C]. In International Conference on Artificial Intelligence and Statistics. 2020 : 2435–2443.
- [16] Lu Keming, Yuan Hongyi, Yuan Zheng et al. # InsTag: Instruction Tagging for Analyzing Supervised Fine-tuning of Large Language Models [C]. In The Twelfth International Conference on Learning Representations. 2023 .
- [17] Hilmkil Agrin, Callh Sebastian, Barbieri Matteo et al. Scaling federated learning for fine-tuning of large language models [C]. In International Conference on Applications of Natural Language to Information

- Systems. 2021 : 15–23.
- [18] Chen Yukang, Qian Shengju, Tang Haotian et al. Longlora: Efficient fine-tuning of long-context large language models [J]. arXiv preprint arXiv:2309.12307. 2023.
- [19] Zhang Haojie, Li Ge, Li Jia et al. Fine-tuning pre-trained language models effectively by optimizing subnetworks adaptively [J]. Advances in Neural Information Processing Systems. 35. 2022: 21442–21454.
- [20] Hu Zhiqiang, Wang Lei, Lan Yihuai et al. Llm-adapters: An adapter family for parameter-efficient fine-tuning of large language models [J]. arXiv preprint arXiv:2304.01933. 2023.
- [21] Brown Tom, Mann Benjamin, Ryder Nick et al. Language models are few-shot learners [J]. Advances in Neural Information Processing Systems. 33. 2020: 1877–1901.
- [22] Malladi Sadhika, Gao Tianyu, Nichani Eshaan et al. Fine-tuning language models with just forward passes [J]. Advances in Neural Information Processing Systems. 36. 2024.
- [23] Laput Gierad P, Dontcheva Mira, Wilensky Gregg et al. Pixeltone: A multimodal interface for image editing [C]. In SIGCHI Conference on Human Factors in Computing Systems. 2013 : 2185–2194.
- [24] Kim Gwanghyun, Kwon Taesung, Ye Jong Chul. Diffusionclip: Text-guided diffusion models for robust image manipulation [C]. In IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022 : 2426–2435.
- [25] Radford Alec, Kim Jong Wook, Hallacy Chris et al. Learning transferable visual models from natural language supervision [C]. In International Conference on Machine Learning. 2021 : 8748–8763.
- [26] Chen Jianbo, Shen Yelong, Gao Jianfeng et al. Language-based image editing with recurrent attentive models [C]. In IEEE Conference on Computer Vision and Pattern Recognition. 2018 : 8721–8729.
- [27] Kawar Bahjat, Zada Shiran, Lang Oran et al. Imagic: Text-based real image editing with diffusion models [C]. In IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2023 : 6007–6017.
- [28] Cheng Yu, Gan Zhe, Li Yitong et al. Sequential attention GAN for interactive image editing [C]. In 28th ACM International Conference on Multimedia. 2020 : 4383–4391.
- [29] Ghodrati Amir, Jia Xu, Pedersoli Marco et al. Towards automatic image editing: Learning to see another you [J]. arXiv preprint arXiv:1511.08446. 2015.
- [30] Brooks Tim, Holynski Aleksander, Efros Alexei A. Instructpix2pix: Learning to follow image editing instructions [C]. In IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2023 : 18392–18402.
- [31] Avrahami Omri, Lischinski Dani, Fried Ohad. Blended diffusion for text-driven editing of natural images [C]. In IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022 : 18208–18218.
- [32] Zhou Yufan, Zhang Ruiyi, Gu Jiuxiang et al. Tigan: Text-based interactive image generation and manipulation [C]. In AAAI Conference on Artificial Intelligence. 2022 : 3580–3588.
- [33] Lee Yao-Chih, Jang Ji-Ze Genevieve, Chen Yi-Ting et al. Shape-aware text-driven layered video editing [C]. In IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2023 : 14317–14326.
- [34] Dong Wenkai, Xue Song, Duan Xiaoyue et al. Prompt tuning inversion for text-driven image editing using diffusion models [C]. In IEEE/CVF International Conference on Computer Vision. 2023 : 7430–7440.
- [35] Li Bo, Lin Xiao, Liu Bin et al. Lightweight text-driven image editing with disentangled content and

- attributes [J]. IEEE Transactions on Multimedia. 2023.
- [36] Rombach Robin, Blattmann Andreas, Lorenz Dominik et al. High-resolution image synthesis with latent diffusion models [C]. In IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022 : 10684–10695.
- [37] Zhang Lvmin, Rao Anyi, Agrawala Maneesh. Adding conditional control to text-to-image diffusion models [C]. In IEEE/CVF International Conference on Computer Vision. 2023 : 3836–3847.
- [38] Vaswani Ashish, Shazeer Noam, Parmar Niki et al. Attention is all you need [J]. Advances in Neural Information Processing Systems. 30. 2017.
- [39] Achiam Josh, Adler Steven, Agarwal Sandhini et al. Gpt-4 technical report [J]. arXiv preprint arXiv:2303.08774. 2023.
- [40] Hu Edward J, Shen Yelong, Wallis Phillip et al. Lora: Low-rank adaptation of large language models [J]. arXiv preprint arXiv:2106.09685. 2021.
- [41] Van Huynh Nguyen, Hoang Dinh Thai, Nguyen Diep N et al. DeepFake: Deep dueling-based deception strategy to defeat reactive jammers [J]. IEEE Transactions on Wireless Communications. 20 (10). 2021: 6898–6914.

## 致 谢

我对李佩佩老师在本论文的研究和写作过程中的悉心指导和无私帮助深感感激，她严谨的学术态度和深厚的专业知识不断激励着我。同时，感谢所有参与讨论和提供宝贵意见的同学和朋友们，感谢北京邮电大学提供的研究平台和资源，这篇论文的完成离不开每一位给予我支持和帮助的人。

# 北京邮电大学

## 本科毕业设计（论文）开题报告

学院	人工智能学院		专业	智能科学与技术	
学生姓名	罗彬慈	学号	2020212053	班级	2020219107
指导教师姓名	李佩佩	所在单位	人工智能学院	职称	副教授
设计（论文）题目	(中文) 基于对话系统的交互式图像编辑的研究与仿真				
	(英文) Research and Simulation of Interactive Image Editing Based on Conversational Systems				

### 1. 背景和意义

随着图像生成技术的不断发展，图像编辑作为其中的关键技术之一，应用广泛，涵盖了媒体娱乐、数字营销和智能医疗等多个领域。然而，传统的图像编辑模型存在着交互性差和生成图像质量受限的问题，迫使我们探索更先进的方法以提高图像生成的质量和用户交互性。通过深度学习和语言大模型的结合，我们有望构建一个创新的交互式图像编辑系统，为图像编辑领域带来新的可能性。

### 2. 研究的基本内容和拟解决的主要问题

#### 内容一：熟悉图像编辑技术和生成模型的相关知识

在这一部分，基本内容是深入了解图像生成领域的相关知识，包括图像编辑技术、生成模型的原理，以及语言大模型在图像编辑中的应用。这为后续交互式图像编辑系统的构建奠定了理论基础和技术背景。

#### 内容二：分析传统图像编辑模型的限制和挑战

在这一部分，基本内容是深入研究传统图像编辑模型，识别其存在的问题，特别关注交互性和生成图像质量的限制，为改进图像编辑技术提供方向。

#### 内容三：设计和构建交互式图像编辑系统

在这一部分，基本内容是明确交互式图像编辑系统的设计目标和实现方式。通过考察的不同参数量的多种语言大模型，选择在性能和负载中取得平衡的大语言模型，然后设计系统的核心算法和交互界面。

#### 内容四：评估系统性能并进行部署

在这一部分，基本内容是从多个纬度评估系统性能，包括生成图像的质量、交互系统对用户请

求的处理能力和用户体验。通过不断的改进和优化，确保系统能够满足需求，提高系统的可用性和易用性。选择合适的方式部署系统，是系统能够提供服务。

### 3. 研究方法及措施

在解决上述问题的过程中，我们将采用以下研究方法和措施：

(1) 深入调研与学习：对深度学习和神经网络的基本原理进行学习，特别关注对抗生成模型和扩散模型等图像生成模型的详细了解。同时，调研语言大模型的最新发展。

(2) 总结与优化：分析当前图像编辑模型的创新之处和局限性，提出改进图像编辑模型的方法，以提高其性能。

(3) 语言大模型应用：学习如何使用语言大模型，并探索其在图像编辑任务中的应用，为系统集成做好准备。

(4) 系统构建与评估：设计并实现一个交互式图像编辑系统，结合深度神经网络的图像生成模型和语言大模型，通过实验评估系统性能。

### 4. 研究工作的步骤与进度

#### 第 1 阶段（2023.11.20 - 2024.12.03）：背景调研与开题准备

开始调研基于深度神经网络的图像生成模型，了解对抗生成模型（GANs）、扩散模型等的基本原理和应用领域。

准备开题报告，明确研究目标和方法。

#### 第 2 阶段（2023.12.04 - 2023.12.17）：图像生成模型深入研究

深入研究复现的图像生成模型，理解其优点和限制，探索其在图像编辑任务中的潜在应用。

#### 第 3 阶段（2023.12.18 - 2023.12.31）：问题定义与解决思路

针对图像编辑任务，提出创新性的问题，明确解决思路。

#### 第 4 阶段（2024.01.01 - 2024.01.14）：实验与性能评估

进行图像编辑模型的实验，收集和分析实验结果，识别性能瓶颈和问题。

学习语言大模型的使用方法，为系统集成做准备。

#### 第 5 阶段（2024.02.26 - 2024.03.10）：语言大模型与系统集成

结合语言大模型的研究，探讨如何将语言大模型与图像编辑模型相结合，构建交互式图像编辑系统。

第 6 阶段（2024.03.11 - 2024.03.24）：实验与系统调优

进行实验，评估交互式图像编辑系统的性能，进行调整和改进。

第 7 阶段（2024.03.25 - 2024.04.07）：文献更新与中期报告准备

更新文献调研，将最新研究成果与自己的工作相结合，准备中期报告。

第 8 阶段（2024.04.08 - 2024.04.19）：论文撰写与最终总结

完善研究和论文撰写，确保项目完善，准备最终论文和答辩。

主要参考文献：

- [1] ACHIAM, Josh, et al. Gpt-4 technical report. arXiv preprint arXiv:2303.08774, 2023.
- [2] FLORIDI, Luciano; CHIRIATTI, Massimo. GPT-3: Its nature, scope, limits, and consequences. *Minds and Machines*, 2020, 30: 681-694.
- [3] ROMBACH, Robin, et al. High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2022. p. 10684-10695.
- [4] HU, Edward J., et al. Lora: Low-rank adaptation of large language models. arXiv preprint arXiv:2106.09685, 2021.
- [5] ZHANG, Lvmin; RAO, Anyi; AGRAWALA, Maneesh. Adding conditional control to text-to-image diffusion models. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. 2023. p. 3836-3847.
- [6] VAN HUYNH, Nguyen, et al. DeepFake: Deep dueling-based deception strategy to defeat reactive jammers. *IEEE Transactions on Wireless Communications*, 2021, 20.10: 6898-6914.

允许进入毕业设计（论文）下一阶段：是 <input type="checkbox"/> 否 <input checked="" type="checkbox"/>		指导教师
日期	年   月   日	签字

注：可根据开题报告的长度加页

# 北京邮电大学

## 本科毕业设计（论文）中期进展情况检查表

学院	人工智能学院		专业	智能科学与技术	
学生姓名	罗彬慈	学号	2020212053	班级	2020219107
指导教师姓名	李佩佩	所在单位	人工智能学院	职称	副教授
设计（论文） 题目	(中文) 基于 LLM 的交互式多模态图像编辑系统的设计与搭建				
	(英文) Design and Construction of Interactive Multimodal Image Editing System Based on LLM				
目前 已 完 成 任 务	<p>目前已完成的工作主要包括背景调研、系统设计、代码实现和系统部署三个方面：</p> <p><b>一、背景调研</b></p> <p>图像生成技术和 LLM (Large Language Models) 都是深度学习领域的研究热点。Stable Diffusion 技术的推出和不断的迭代引发了在图像生成领域的浪潮；同时，像 ChatGPT 这样的大语言模型的发布和不断升级也引起了全球范围内的广泛关注。</p> <p>图像编辑是图像生成技术中的关键技术之一，其旨在保留图像的主要信息的同时根据指令对特定内容进行修改。图像编辑技术在媒体娱乐、数字营销、智能医疗等领域具有广泛的应用前景，因此备受关注。然而，传统的图像编辑模型在交互性方面存在一定的局限性。本项目旨在利用大语言模型辅助进行交互式图像编辑，结合先进的图像生成模型以提升生成图像的质量，最终打造一个交互式图像编辑系统。</p> <p><b>二、系统设计</b></p> <p>1. 系统框架</p> <p>考虑到本项目需要整合多个模型且要满足用户使用的便利性，本项目使用 Golang 语言搭建了一个后端服务（middleware）对相同类型功能的请求进行整合并向 GUI 提供整合后的 API，因此本项目的 GUI 部分仅需安装一个 python 依赖库，降低了使用门槛。图像生成模型的请求通过 API 调用部署在云平台上的 Stable Diffusion 模型和由 OpenAI 提供的 DALL-E 模型；大语言模型的调用目前可选择通过 API 调用 OpenAI 的 GPT3.5turbo/GPT4 模型，后续准备对 ChatGLM 系列模型进行微调后部署在云平台上以通过 API 调用。这样的系统框架增加了用户使用的便捷性、系统迭代的稳定性，可在用户无感的条件下对系统进行升级。</p> <pre style="background-color: #f0f0f0; padding: 10px;"> <b>gradio_web(Image preprocessing, GUI)</b>           api       ---api---&gt; StableDiffusion           middleware &lt;== ---api---&gt; OpenAI                                 (GPT3.5turbo/GPT4, DALLE)  ---api---&gt; ChatGLM2-6B </pre>				

图 2-1 系统框架

	<p><b>2.大语言模型</b></p> <p>大语言模型主要通过使用给定的 Prompt、图像分割的结果等将用户输入的自然语言转换为 json 格式的指令。系统会在大语言模型返回的文本中自动地抽取指令并结合配置文件提供的规则对指令的合规性进行校验。由于不同的语言大模型的表现并不一致，用户可以在配置文件中添加多个不同的 Prompt 模版以在各个模型上达到更好的效果。</p>
	<p><b>3.图像生成模型</b></p> <p>考虑到 Stable Diffusion 的功能更加丰富以及在开源社区的热度，图像生成主要采用 Stable Diffusion 模型，并在特定的任务中结合不同的模型与插件以满足任务的需求并提高结果的质量。当用户上传图片时，图像分割模型会对图像进行分割，并将分割的结果以对话的方式反馈给用户。在使用图像生成模型对图像进行修改时，首先会结合大语言模型生成的指令和分割的结果对不需要修改的部分进行遮罩，然后在生成的过程中通过 Control Net 对图像的基本框架进行固定以保证生成的质量。由于图像分割模型返回的分割结果不可避免的在一些细节上存在瑕疵（如标签 Hair 在部分区域存在将发丝周围的少量的像素一同分割），在生成遮罩时会对特定的标签进行不同程度的腐蚀--膨胀操作以提高遮罩的质量。同时，可以选择是否使用 ROOP 对图像中的人脸进行替换。当部署在云平台上的 Stable Diffusion 模型不可用时，会使用 DALL-E 模型完成相应功能。</p>
	<p><b>三、代码实现</b></p> <p>目前的总代码量为 3649 行，其中 GUI 代码量为 2323 行， middleware 代码量为 1120 行。</p> <p><b>1.GUI</b></p> <p>使用 Python3 实现了 GUI 的构建，主要使用了 gradio 库。在代码实现时将 GUI 分为 webui、controllers、modules、config、test 五个模块，分别实现 UI 布局（各个模块在 UI 界面的分布）、交互逻辑（不同可交互模块的处理逻辑）、请求处理（图像处理、request body 生成等）、系统配置（模版、路由、默认参数等）、代码测试（主要针对 modules 中的函数进行测试）。UI 界面主要由操作面板、图像预览、文本交互界面和图片编辑界面。</p> <p><b>2.middleware</b></p> <p>使用 Golang 语言配合 beego 框架实现，将大语言模型调用、图像生成模型调用等数十个 API 整合为 8 个 API 供 GUI 调用，同时在不需要标识请求用户的情况下支持多个 GUI 同时对其进行访问但互不影响（如对话的历史记录等）。</p> <p><b>3.Stable Diffusion</b></p> <p>基于开源项目 stable-diffusion-webui 进行修改，在其中添加了 ROOP、Control Net 等插件以满足项目的需求。</p>
	<p><b>四、系统部署</b></p> <p><b>1.项目运行</b></p> <p>本项目提供了 Docker、Kubernetes、Terminal 等多种运行方式。</p> <p><b>2.持续集成、持续部署</b></p> <p>持续集成（Continuous Integration, CI）是一种软件开发实践，旨在通过频繁地将代码集成到共享存储库中，然后对代码进行自动化构建和测试，以确保代码变化不会破坏整个代码库的稳定性。CI 旨在尽早发现和解决集成问题，从而提高开发效率和软件质量。</p> <p>持续部署（Continuous Deployment, CD）持续部署是自动将经过测试的代码部署到生产环境，没有人工干预。</p> <p>本项目使用 GitHub Action 实现了自动化测试、镜像自动化构建与发布、自动部署至 Azure，提高了代码的可靠性并减少了非开发的工作量。</p>
是否符合任务书要求进度	是 <input type="checkbox"/> 否 <input type="checkbox"/>

尚需完成的任务	<p>1. 实现建议功能，在对话时提供建议，并且在用户采纳建议后更新指令。由于传统的大语言模型只能接收文本输入，如果使用传统的大语言模型效果不理想，考虑使用 GPT4-V 模型来生成更好的建议。</p> <p>2. 使用 LoRa 微调一个更加适合本任务的大语言模型，在与大语言模型交互时支持切换不同的大语言模型。</p>		
	是否可以按期完成设计（论文） 是 <input type="checkbox"/> 否 <input type="checkbox"/>		
存在问题和解决办法	存在问题	<p>1. GUI 的交互性仍有待改善。</p> <p>2. 在编辑图像时可选的参数较少。</p> <p>3. 图像生成质量不稳定。</p>	
	拟采取的办法	<p>1. 通过对功能分区分类或简化提升 GUI 的交互性。</p> <p>2. 实现更多的参数可自定义并增加一个设置面板，提高系图像生成的稳定性，将所有设置内容迁移到设置面板中并提供模版，避免在增加参数的过程中增加交互的复杂性。</p>	
指导教师签字		日期	年   月   日
检查小组评分及意见	<p>评分： (总分： )</p> <p style="text-align: right;">组长签字： 年   月   日</p>		

注：此表仅供参考，各学院应围绕毕设目标达成度，结合人才培养目标、专业认证要求等进行个性化完善。

北 京 邮 电 大 学

教师指导本科毕业设计（论文）记录表

学院	专业	班级	
学生姓名	学号	班内序号	
指导教师姓名	职称		
第 1—2 周记录:			
指导教师签字		日期	年 月 日
第 3—4 周记录:			
指导教师签字		日期	年 月 日

注：每 2 周指导内容记录在一个表格中，双面打印。