



# 基于生成式 AI 的个性化文创图像作品设计系统 需求分析



任课教师 \_\_\_\_\_ 杨波

学 院 \_\_\_\_\_ 计算机学院

专 业 \_\_\_\_\_ 计算机科学与技术

组 别 \_\_\_\_\_ 第一组

组 长 \_\_\_\_\_ 郑仕博

成 员 \_\_\_\_\_ 陈奕嘉，苏泳豪

2025 年 6 月 3 日

# 目录

<b>1</b>	<b>系统概述</b>	<b>2</b>
1.1	产品目标与愿景 . . . . .	2
1.2	目标用户 . . . . .	2
1.3	产品范围 . . . . .	2
1.4	假设与依赖 . . . . .	2
<b>2</b>	<b>总体设计</b>	<b>2</b>
2.1	系统架构 . . . . .	2
2.2	系统组成 . . . . .	3
2.3	数据设计 . . . . .	3
<b>3</b>	<b>功能需求</b>	<b>4</b>
<b>4</b>	<b>性能指标</b>	<b>5</b>
<b>5</b>	<b>接口需求</b>	<b>6</b>
<b>6</b>	<b>其它要求</b>	<b>6</b>
6.1	安全性、可维护性等要求 . . . . .	6
6.2	运行环境要求 . . . . .	6
<b>7</b>	<b>技术难点</b>	<b>6</b>
<b>8</b>	<b>附录:</b>	<b>7</b>
<b>9</b>	<b>针对 AI 产品的额外考虑:</b>	<b>7</b>
<b>10</b>	<b>总结:</b>	<b>7</b>

# 1 系统概述

## 1.1 产品目标与愿景

本产品的核心目标为构建一个基于生成式 AI 的文创产品生成系统，旨在解决目前人们对文创产品的审美需求与设计需求，未来将考虑更加个性化的生成用于服务用户需求。

## 1.2 目标用户

本产品的目标用户群体为文创产品爱好者以及文创产品设计工作者，这些用户具有较高的审美门槛。

## 1.3 产品范围

本产品目前的功能为通过用户合理输入生成符合中华人民共和国宪法法律的相关文创图片，不用于生成成人、政治敏感、违法犯罪内容，不用于生成视频。

## 1.4 假设与依赖

本项目假设用户将合法使用本产品，产品运行需要相关服务器，相关接口，相关数据集。

# 2 总体设计

## 2.1 系统架构

- 用户界面
  - 用户通过浏览器或应用程序客户端与系统进行交互。
  - 提供文字输入框、图片上传功能、文字位置和样式标注工具、文创风格选择器、生成按钮、结果预览区域以及保存和分享功能。
  - 使用现代 Web 前端技术栈（Gradio）构建。
- 后端 API
  - 接收来自前端的请求，处理业务逻辑，并协调各个服务模块。
  - 使用 Python (例如：Gradio) 开发。
- AI 模型服务
  - 部署和运行核心的 AI 模型，负责根据用户输入生成创意图片。

- 可以进行图文融合。
- 包含文本编码器用于理解文字输入。
- 需要强大的计算资源来支持模型推理。

## 2.2 系统组成

### • 软件组成

- **前端应用**: 使用 Gradio 库构建的 Web 界面。
- **后端 API 服务**: 基于 Python (Gradio 或其他框架) 开发的。
- **AI 模型**:
  - \* **基础模型**: 类似 Stable Diffusion 1.5 的预训练扩散模型。
  - \* **微调模型 (可选)**: 基于特定文创风格或数据集进行微调的模型, 以生成更符合特定需求的图片。
  - \* **文本编码器**: 用于将用户输入的文字转换为模型可以理解的特征向量。
- **依赖库**:
  - \* **Gradio**: 用于快速构建用户界面和 API。
  - \* **PyTorch 或 TensorFlow**: 用于加载和运行 AI 模型。
  - \* **Transformers 库 (Hugging Face)**: 提供预训练模型和相关工具。
  - \* **Pillow (PIL)**: 用于图像处理。

### • 硬件组成

- **服务器**:
  - \* **API 服务器**: 运行后端 API 服务的服务器, 需要具备一定的 CPU 和内存资源。
  - \* **AI 模型服务器**: 搭载高性能 GPU (例如: NVIDIA RTX 系列或更高) 的服务器, 用于加速 AI 模型的推理计算。根据预期用户量和性能需求, 可能需要多台 GPU 服务器。
- **用户终端**: 用户使用的电脑、平板电脑或手机等设备, 只需要能够运行现代 Web 浏览器即可访问前端界面。

## 2.3 数据设计

### • 模型数据:

- **模型权重文件**: 与 AnyText 模型结构相同的权重文件。

- **模型配置文件:** 存储模型结构、超参数等信息的配置文件。
- **生成图片数据:**
  - **生成的文创图片文件:** 存储实际生成的图片文件 (例如: .png 或.jpg 格式)。
  - **图片元数据:** 记录生成图片的相关信息, 例如
    - \* 生成数间
    - \* 使用的模型版本
    - \* 对应的用户输入参数
    - \* 分辨率
    - \* 文件大小

### 3 功能需求

#### 方式一: 基于用户输入的文字

1. **文字输入:** 用户可以在界面上提供的文本输入框中键入或粘贴想要使用的文字内容, 例如一句诗词、一句格言、一个祝福语、或者任何简短的文本。
2. **供参考的物品:** 用户可以在界面上看到所有进行加强训练的物品, 模型在生成这些物品的准确率上有一定的保证, 同时也可以看到比较优美的一些例子, 供用户参考。
3. **文字位置标注:**
  - 自由拖拽模式: 用户可以使用鼠标在预览区域内拖拽一个矩形框, 以指定文字将要放置的位置和大致大小。
  - 预设位置选择: 系统可以提供一些常用的预设位置选项 (例如: 居中、左上角、右下角等), 供用户快速选择。
4. **图片生成:** 用户点击“生成”按钮后, AI 产品将根据用户输入的文字、标注的位置、选择的文字样式和文创风格, 生成一张带有文字的创意图片。
5. **结果预览与调整:** 生成的图片会显示在预览区域, 用户可以查看效果。如果对结果不满意, 可以返回修改文字内容、位置、样式或风格, 然后重新生成。
6. **保存与分享:** 用户可以将生成的文创图片保存到本地设备, 或者分享到社交媒体平台。

#### 方式二: 基于用户输入的成型图片

1. **图片上传:** 用户可以通过文件上传功能上传一张已经存在的图片作为创作的基础。支持常见的图片格式，例如 JPG、PNG 等。
2. **供参考的物品:** 用户可以看到一些比较优美的例子，但仅供参考。
3. **文字位置标注:**
  - 与方式一类似，用户可以使用自由拖拽模式或预设位置选择在上传的图片上标注文字将要放置的位置和大小。
  - 用户可以添加多个文本框，标注多个文字的位置。
4. **文字内容输入与样式定制:** 对于每个标注的位置，用户可以输入相应的文字内容。
5. **图片生成:** 用户点击“生成”按钮后，AI 产品将在用户上传的图片上，根据标注的文字位置和样式，以及添加的其他文创元素，生成最终的文创图片。
6. **结果预览与调整:** 生成的图片会显示在预览区域，用户可以查看效果。用户可以调整文字的位置、样式，或者添加/修改其他文创元素，然后重新生成。
7. **保存与分享:** 用户可以将最终生成的文创图片保存到本地设备，或者分享到社交媒体平台。

总而言之，该 AI 产品旨在提供一个简单易用、功能丰富的界面，让用户能够通过输入文字或上传图片，并灵活地标注文字的位置和定制样式，最终生成具有个性化创意的文创图片。

## 4 性能指标

本系统从文字生成与编辑两个方面对模型性能进行评估。

- **文字生成与编辑评估:** 采用 OCR 模型对生成图像中的文字进行识别，提取生成文本后与真实文本进行比对，计算准确率（仅完全匹配计为正确）与编辑距离，以衡量系统在控制文字内容与位置方面的精度。
- **生成图像质量评估:** 对于图像整体质量，使用 FID (Fréchet Inception Distance) 指标对 SD1.5 扩散模型生成的图像进行评估，并与原有模型进行对比，验证微调是否保留了原有模型的图像生成能力，避免过度遗忘 (catastrophic forgetting) 现象。
- **基准测试工具:** 所有测试均采用 AnyText 官方发布的 AnyText-benchmark 工具链完成，确保结果的客观性和可比性。

## 5 接口需求

- 应用程序接口 (API) 需求: 并不调用其它应用程序接口。
- 数据接口需求: **Gradio** 本身主要负责用户界面的展示和与后端 **Python** 代码的交互, 它不直接处理与外部数据源的连接。
- 硬件接口需求: 本项目不涉及硬件接口。

## 6 其它要求

### 6.1 安全性、可维护性等要求

- 安全性:
  - 数据隐私保护: 确保对话发生在本地, 不收集用户数据。
  - 防止恶意攻击: 强化防御体系, 防止恶意攻击。
  - 访问控制: 对于进行过恶意访问的 **IP** 进行拉黑, 减少恶意访问量。
- 可维护性:
  - 代码的可读性与可扩展性较好, 详细注释了大部分功能。
  - 代码易于部署, 容易维护。

### 6.2 运行环境要求

1. 硬件要求: GPU 4060, 8GB 显存, 32GB 内存。
2. 环境要求: 详见 [https://github.com/SPM-PSP/PSP25-1/blob/main/codes/front\\_back\\_ends/environment.yaml](https://github.com/SPM-PSP/PSP25-1/blob/main/codes/front_back_ends/environment.yaml)
3. 网络要求: 需要连接互联网以下载模型和数据集。

## 7 技术难点

在本系统的研发过程中, 我们面临如下几个关键技术难点:

- **多语言文字嵌入与位置对齐困难:** 文字控制框架要求在图像中精确嵌入特定语言 (尤其是中文) 文字, 同时控制其空间位置和字体风格。由于中文字符集庞大、形态复杂, 构造稳定有效的嵌入表示并保证其与图像特征对齐是一大挑战。

- **文字生成与编辑一致性不足:** 文字编辑功能需对图像局部区域进行内容修改, 保持背景纹理一致且不引入伪影。实现高保真的图文融合效果, 要求模型不仅生成文字形状, 还能恢复修改区域的上下文信息, 对网络结构和训练策略提出更高要求。
- **训练资源消耗大, 优化困难:** 文字控制部分使用大规模数据 (40 万图文对), 扩散模型训练需使用 DreamBooth 微调方法, 均对显存和计算资源要求极高。由于训练轮数和数据吞吐量受限, 实际训练效果存在瓶颈, 尤其在准确率和泛化能力方面表现不理想。
- **评估方法易受 OCR 误差干扰:** 系统主要通过 OCR 提取文字用于评估生成准确率和编辑距离。然而 OCR 本身存在识别误差, 会导致准确率评估不稳定, 尤其在文字模糊或艺术字体场景下, 难以反映真实效果。
- **图像扩散模型易遗忘原始生成能力:** 扩散模型微调 (如 Realistic\_Vision\_V4.0) 在少量样本上优化文字控制能力时, 容易遗忘已有的视觉先验 (catastrophic forgetting), 影响模型的基础图像生成质量。如何在增强文字能力的同时保持图像多样性与清晰度, 是核心难题之一。

## 8 附录:

- **A 术语与缩略语:**

**AnyText:** 发布的一种模型, 对文字渲染进行了强化训练。

**sd1.5:**Stable Diffusion v1-5。

**Gradio:**Gradio 是一个开源的 Python 库, 用于快速构建机器学习和数据科学模型的交互式 Web 界面。

## 9 针对 AI 产品的额外考虑:

- **数据需求:** 文字和图片数据
- **模型部署和监控:** 在满足上述要求的服务器上, 运行脚本进行部署, 远程连接 (todesk) 可进行实时监控。

## 10 总结:

本产品是一个基于生成式 AI 的文创图片生成系统, 旨在帮助文创爱好者和设计者通过输入文字或上传图片并标注文字位置, 快速生成创意图片。系统采用 Gradio 构建前后端, 使用类似 Stable Diffusion 1.5 的 AI 模型进行图文融合。部署



需要 GPU 服务器。用户可以输入文字或上传图片进行创作，并可对文字位置和样式进行调整，最终保存和分享作品。