

基于生成式生成式 AI 的个性 化文创图像作品设计系统 使用手册

目录

文 件 修 订 记 录 2

1. 总体功能描述 3

2. 运行环境 5

3. 软件编译环境 6

4. 软件使用说明 7

5. 图像存储 11

6. Docker 镜像部署 12

7. 规范文件 13

文件修订记录

版本号	生成日期	作者	修订内容
V1.0	2025.4.10	郑仕博	初始版本

1. 总体功能描述

系统模型采用三部分构成，分别是 Text-control Diffusion Pipeline、Auxiliary Latent Module、Text Embedding Module，并采用深度学习常用的 Python 语言和 Gradio 开发框架进行开发。

使用方式如下：

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，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：基于用户输入的成形图片

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, 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。

(需要渲染的文字需要用双引号标注, 全角半角均可以, 例如: 太阳神鸟金饰上写着“天路”。从而渲染天路这两个文字)

2. 运行环境

硬件要求

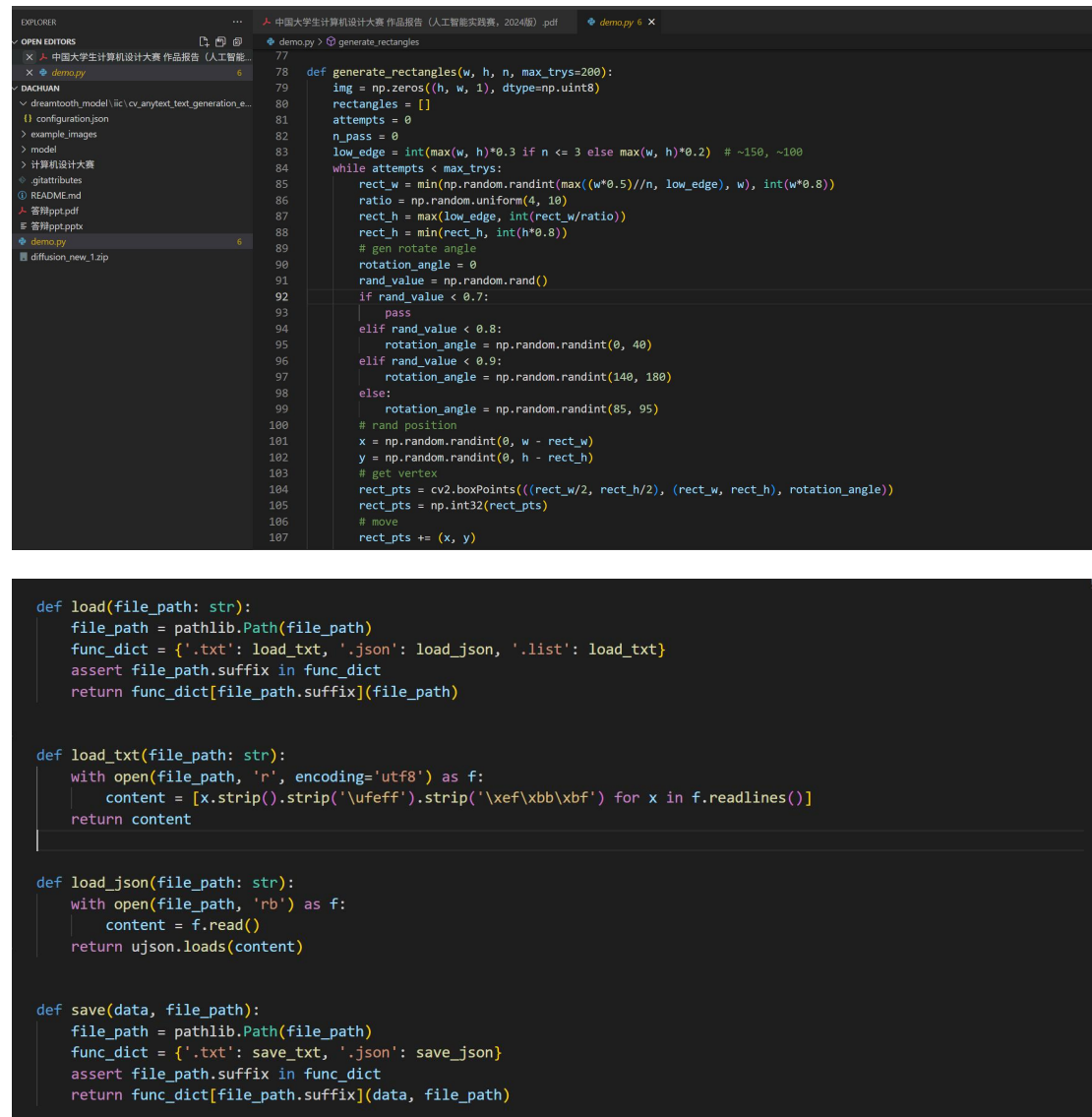
类 别	基本要求
服务器端	内存 32G 以上; GPU NVIDIA RTX 4060 内存 8G 以上; 硬盘剩余空间不低于 50G;
客户端	能运行现代网页浏览器即可

软件要求:

类别	名 称	基本环境
服务器端	操作系统	支持 Docker 即可
	Docker	要求下载 Docker 以及 Docker 的 Nvidia 驱动
客户端	操作系统	任意操作系统
	其它软件	浏览器 (Google, Edge 等)

3. 软件编译环境

本软件使用 Visual Studio Code 进行开发，使用 Python 和 Javascript 进行开发编译。



```

77
78 def generate_rectangles(w, h, n, max_trys=200):
79     img = np.zeros((h, w, 1), dtype=np.uint8)
80     rectangles = []
81     attempts = 0
82     n_pass = 0
83     low_edge = int(max(w, h)*0.3 if n <= 3 else max(w, h)*0.2) # ~150, ~100
84     while attempts < max_trys:
85         rect_w = min(np.random.randint(max((w*0.5)//n, low_edge), w), int(w*0.8))
86         ratio = np.random.uniform(4, 10)
87         rect_h = max(low_edge, int(rect_w/ratio))
88         rect_h = min(rect_h, int(h*0.8))
89         # gen rotate angle
90         rotation_angle = 0
91         rand_value = np.random.rand()
92         if rand_value < 0.7:
93             pass
94         elif rand_value < 0.8:
95             rotation_angle = np.random.randint(0, 40)
96         elif rand_value < 0.9:
97             rotation_angle = np.random.randint(140, 180)
98         else:
99             rotation_angle = np.random.randint(85, 95)
100         # rand position
101         x = np.random.randint(0, w - rect_w)
102         y = np.random.randint(0, h - rect_h)
103         # get vertex
104         rect_pts = cv2.boxPoints(((rect_w/2, rect_h/2), (rect_w, rect_h), rotation_angle))
105         rect_pts = np.int32(rect_pts)
106         # move
107         rect_pts += (x, y)

def load(file_path: str):
    file_path = pathlib.Path(file_path)
    func_dict = {'txt': load_txt, '.json': load_json, '.list': load_txt}
    assert file_path.suffix in func_dict
    return func_dict[file_path.suffix](file_path)

def load_txt(file_path: str):
    with open(file_path, 'r', encoding='utf8') as f:
        content = [x.strip().strip('\uffff').strip('\xef\xbb\xbf') for x in f.readlines()]
    return content

def load_json(file_path: str):
    with open(file_path, 'rb') as f:
        content = f.read()
    return ujson.loads(content)

def save(data, file_path):
    file_path = pathlib.Path(file_path)
    func_dict = {'txt': save_txt, '.json': save_json}
    assert file_path.suffix in func_dict
    return func_dict[file_path.suffix](data, file_path)
    
```

4. 软件使用说明

本软件将发布至 Web 端，可以在浏览器中输入相关网址或名称，然后点击使用。

打开将会展示说明界面以及功能界面，效果如图：



模型路径:

Parameters(参数)

Base Model Path(基模地址)

LoRA Path and Ratio(lora地址和比例)

Prompt(提示词)

Text Generation(文字生成)

Text Editing(文字编辑)

Pos-Method(位置方式)

choose a method to specify text positions(选择方法用于指定文字位置).

Manual-draw(手绘)

Manual-rect(拖框)

Auto-rand(随机)

Sort Position(位置排序)

position sorting priority(位置排序时的优先级)

↕

↔

Revise Position(修正位置)

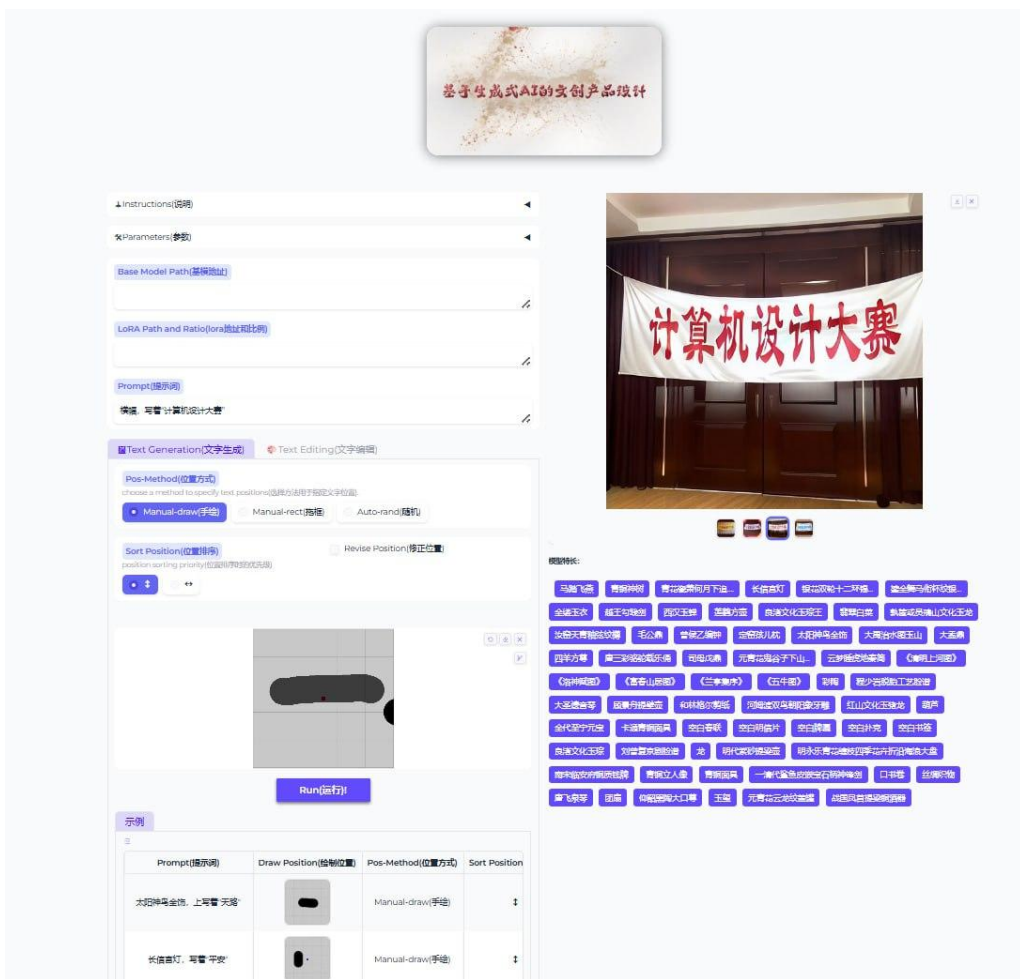
Start Drawing

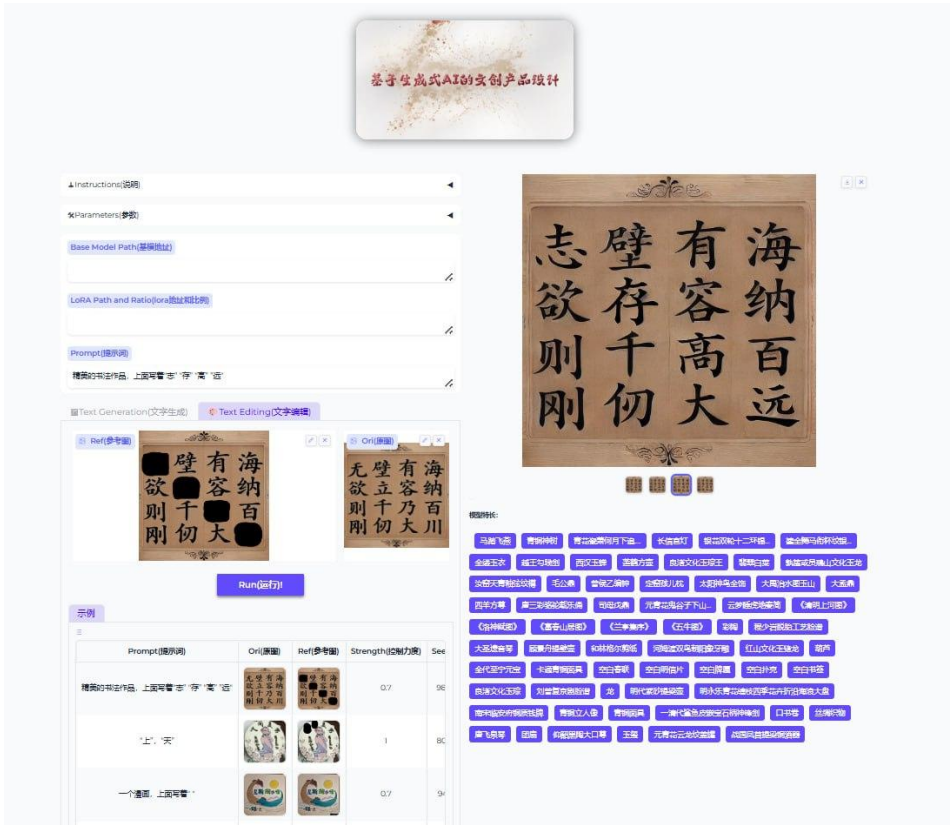
Run(运行!)

示例

Prompt(提示词)	Draw Position(绘制位置)	Pos-Method(位置方式)	Sort Position(位置排序)
太阳神鸟金饰, 上写着"天路"		Manual-draw(手绘)	↕
长信宫灯, 写着"平安"		Manual-draw(手绘)	↕
水晶白菜, "白菜"		Manual-draw(手绘)	↕
卡通青铜树, 上方写着"神树"		Manual-draw(手绘)	↕
卡通青铜面具, 头顶刻着"王"		Manual-draw(手绘)	↔

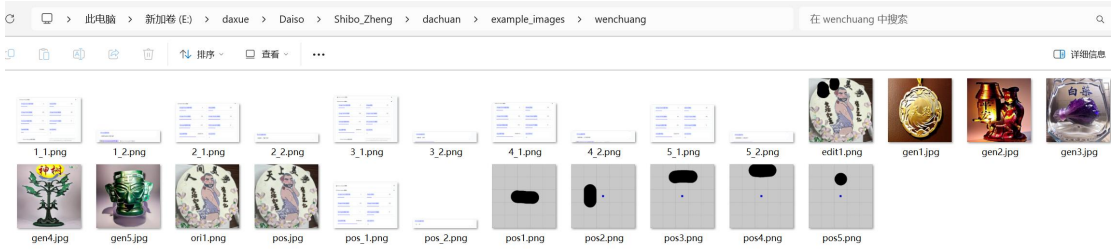
8





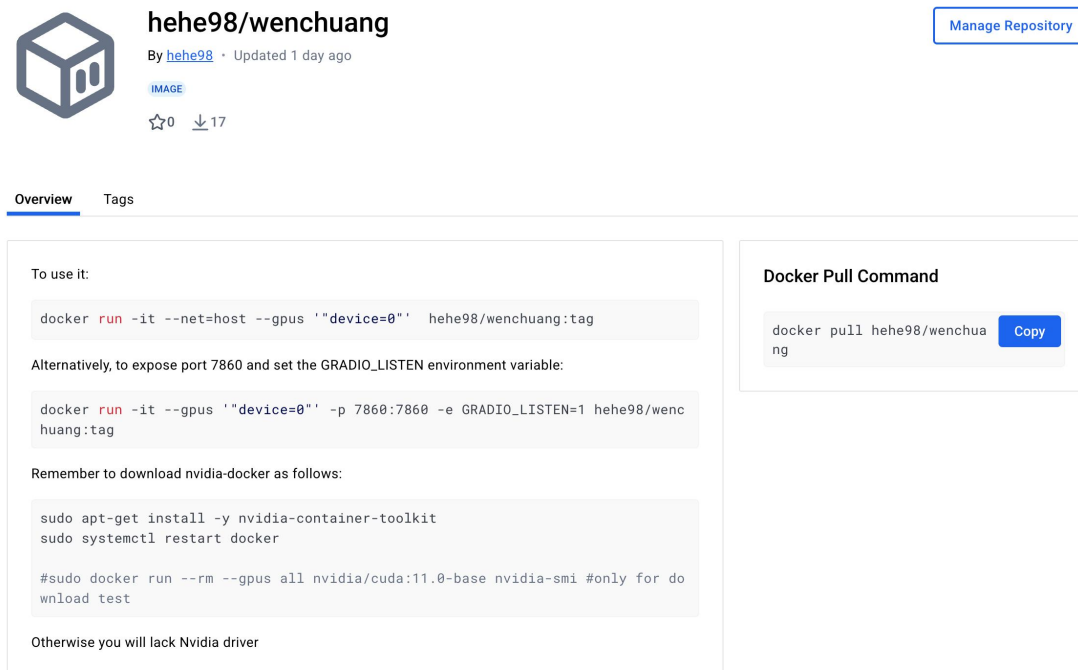
5. 图像存储


待图片生成成功后，后端会将图片存储至指定文件夹中，效果如图：



6. Docker 镜像部署

本软件将项目包装成 docker image 并上传至 Docker Hub，使用 `docker pull hehe98/wenchuang:tag`，按照说明运行容器即可，如下：



 **hehe98/wenchuang** [Manage Repository](#)

By [hehe98](#) · Updated 1 day ago

IMAGE

☆0 ↓17

[Overview](#) [Tags](#)

To use it:

```
docker run -it --net=host --gpus '"device=0"' hehe98/wenchuang:tag
```

Alternatively, to expose port 7860 and set the GRADIO_LISTEN environment variable:

```
docker run -it --gpus '"device=0"' -p 7860:7860 -e GRADIO_LISTEN=1 hehe98/wenchuang:tag
```

Remember to download nvidia-docker as follows:

```
sudo apt-get install -y nvidia-container-toolkit
sudo systemctl restart docker

#sudo docker run --rm --gpus all nvidia/cuda:11.0-base nvidia-smi #only for do
wnload test
```

Otherwise you will lack Nvidia driver

Docker Pull Command

```
docker pull hehe98/wenchuang
```

[Copy](#)

7. 规范文件

详见提交材料中的“项目注意事项”文档