

# 画风迁移系统测试报告

## 1.引言

### 1.1 目的

本测试分析报告为画风迁移软件的测试分析报告，目的在于总结测试阶段的测试以及分析测试的结果，描述系统是否符合需求。

### 1.2 背景

被测试软件：画风迁移。测试环境与实际运行环境同为一般 PC，配置为通用配置。

## 2.测试概要

### 2.1 测试环境

	机器类型	硬件配置	操作系统	其他应用软件
客户端	笔记本电脑	显卡：NVidia 1050 内存：8G	Windows	Spyder Anaconda

由于本软件的结构较为简单，本次测试采用对同一张图片应用不同画风的方法

### 2.2 测试组织

角色	姓名	具体职责
测试管理员	李源钊、黄京津	测试策划：安排测试人员进行测试，准备客户终端 测试设计：根据需求规格说明书的内容对系统功能进行检验和测试 测试总结：代码健壮性较强，可维护性较强。
测试人员	张忠宇、张拓、李卓航	测试执行：记录测试过程和结果，并进行分析

2.3 测试类型

测试类型	测试阶段		
	单元测试	集成测试	系统测试
功能测试	路径正常	无与接口相关的错误	产品的功能符合系统需求
性能测试			性能良好
兼容性测试		能在 windows 系统上顺利运行	

3.测试结果及分析

输入数据		预期输出结果	运行输出结果	结果正常	
				是	否
正常测试输入数据	Jpg 格式图片	修改画风后的图片	拥有新画风的图片	✓	
	Png 格式图片	修改画风后的图片	拥有新画风的图片	✓	
导致反常输入数据	Gif 动态图片	无法执行该文件	文件执行失败	✓	
	非图片文件	无法执行该文件	文件执行失败	✓	

3.1 基本功能测试

如图 1 以及图 2，分别是输入原图以及想要迁移的风格图。



图 1 输入原图



图 2 目标风格图

有了以上两个图片，经过画风迁移系统得到的目标图如图 3（300 个 epoch 后生成图）：

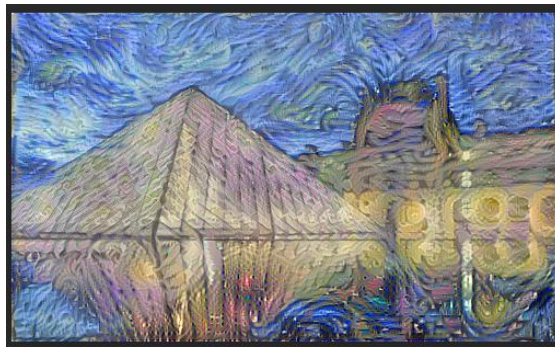


图 3 生成的目标图

由图 3，可以看出生成的目标图已经具备了输入图片的大小以及内容，同时还含有迁移图片的画风，效果不太好的原因是因为，训练代数可能过少。

### 3.2 启动代码测试

```
if __name__ == "__main__":  
    setup()  
    # 指定图片  
    content_img = "contents/sky.jpg"  
    style_img = "styles/starry_night.jpg"  
    # 指定像素尺寸  
    img_width = 400  
    img_height = 300  
    # style transfer  
    style_transfer = StyleTransfer(content_img, style_img, img_width, img_height)  
    style_transfer.build()  
    style_transfer.train(300)
```

## 4.对软件功能的结论

画风迁移软件能较好地完成图片的画风改变，神经网络的运用使得软件的性能得到了很大程度的提升，满足了系统需求，进一步满足了用户的使用需求。

单一的画风迁移功能也使得软件运行的错误率降到极低，在硬件设施提升的基础上可以考虑软件功能的进一步发展。