

Pix2Answer: 数学公式图像识别与转换系统

所有成员：井旺波 邱天爽 魏一凡 张显 周根发

指导老师：陈欢欢

中国科学技术大学2023级011系

摘要： Pix2Answer 是一个创新的软件解决方案，旨在解决数学领域中图像内公式识别与转换的挑战。该系统利用先进的光学字符识别（OCR）技术和 LaTeX 公式处理算法，能够从上传的图像中精确识别数学公式，并将其转换为 LaTeX 格式，进而进行数学运算和结果可视化。系统还配备了一个直观的图形用户界面（GUI），使用户能够轻松上传图片、查看识别的公式、计算结果，以及处理后的图像预览。

关键词： 数学公式识别，OCR，LaTeX，图形用户界面，计算结果可视化

1 项目背景

1.1 选择该项目的原因

数学公式在学术研究、教育和工程实践中占有重要地位，然而手动输入公式不仅耗时且容易出错。现有的图像识别技术虽然已经有了一定的发展，但在数学公式识别和处理方面仍存在诸多挑战。因此，我们决定开发 Pix2Answer，以期提高数学公式输入和处理的效率和准确性。

1.2 相关的科学与社会背景

随着人工智能和光学字符识别技术的发展，自动化处理图像中的文本信息已经成为可能。然而，数学公式的复杂性和多样性使得识别和处理这类图像具有较高的技术难度。本项目旨在结合最新的OCR技术和数学公式处理算法，提供一个高效、准确的解决方案。

2 项目目标

Pix2Answer 项目的主要目的是开发一个能够识别和处理图像中数学公式的系统。具体目标包括：

- 实现高精度的数学公式识别。
- 将识别的公式转换为标准的 LaTeX 格式。
- 对 LaTeX 公式进行数学运算，并生成相应的结果图像。
- 提供用户友好的图形界面，方便用户操作。

3 项目实施过程

3.1 需求分析

在项目初期，我们通过调查问卷和访谈方式，收集了潜在用户对数学公式识别软件的需求。这些需求主要集中在识别准确性、处理速度、用户界面友好性等方面。

3.2 技术选型

经过多次讨论和调研，我们选择了基于卷积神经网络的OCR技术和LaTeX公式处理算法作为核心技术，并使用Python作为主要开发语言，结合tkinter进行界面开发。

3.3 设计与实践

我们设计了系统的整体架构，包括图像预处理模块、公式识别模块、LaTeX转换模块和用户界面模块。在开发过程中，我们采用了迭代开发的方法，不断进行功能添加和优化。

3.4 测试与优化

在系统开发完成后，我们进行了多轮内部测试和用户测试，针对测试中发现的问题进行了优化和改进，最终实现了系统的稳定和高效运行。

4 项目内容

4.1 项目概述

Pix2Answer 是一个大学生小组作业项目，旨在识别图片中的数学公式并计算结果。项目主要使用了 `pix2tex`、`latex2sympy2` 和 `matplotlib` 等库，并基于 `tkinter` 创建了图形用户界面。

4.2 项目结构

- `.gitignore`: 配置文件，用于忽略不需要纳入版本控制的文件和目录。
- `README.md`: 项目介绍和作者信息。
- `main.py`: 主程序文件，包含图像处理和 GUI 初始化的逻辑。
- `module.py`: 包含数学公式转换和图像生成的功能模块。
- `test.py`: 测试文件，用于验证图像识别和公式计算功能。
- `view.py`: GUI 视图模块，定义了用户界面的各个组件和功能。
- `log.txt`: 记录项目开发过程的日志文件

4.3 主要功能

4.3.1 图像处理和公式识别:

1. 使用 PIL 库打开并处理图像。
2. 使用 `pix2tex` 的 `LatexOCR` 类进行公式识别。
3. 使用 `latex2sympy2` 库将识别到的 LaTeX 公式转换为 `sympy` 公式，并进行计算。

4.3.2 LaTeX 公式转换和图像生成:

1. 使用 `latex2latex` 函数将字符串输入转换为标准的 LaTeX 输出。
2. 使用 `matplotlib` 库生成 LaTeX 公式对应的图像。
3. 通过临时文件保存和删除中间图像文件，确保内存的有效使用。

4.3.3 图形用户界面：

1. 使用 `tkinter` 创建了简单易用的 GUI 界面，用户可以通过界面上上传图片、显示识别的 `LaTeX` 公式及其计算结果。
2. 支持显示处理后的外部图像，并提供重新启动应用程序的功能。

4.4 详细日志

1. 项目初始化

- 2024-05-22 15:35:53.850 TRACE [MainThread] [DEBUG] - Project::Initialization - 创建 `.gitignore` 文件，配置忽略规则，确保版本控制系统不跟踪无关文件。

2. 编写 README.md

- 2024-05-22 16:00:12.567 TRACE [MainThread] [DEBUG] - Documentation::README - 完成项目介绍、使用库和作者信息的编写。

3. 实现主要功能模块

- `main.py` :
 - 2024-05-23 09:12:45.123 TRACE [MainThread] [DEBUG] - Main::Imports - 导入所需模块和库，定义图像处理和 GUI 初始化逻辑。
 - 2024-05-23 09:30:22.456 TRACE [MainThread] [DEBUG] - Main::ProcessImage - 编写 `process_image` 函数，实现图像处理和公式识别。
 - 2024-05-23 10:15:33.789 TRACE [MainThread] [DEBUG] - Main::MainFunction - 实现 `main` 函数，初始化并运行 `tkinter` 主窗口。
- `module.py` :
 - 2024-05-23 11:05:44.000 TRACE [MainThread] [DEBUG] - Module::Pix2Answer - 实现 `pix2answer` 函数，将识别到的像素值转换为标准 `LaTeX` 公式。
 - 2024-05-23 11:20:15.567 TRACE [MainThread] [DEBUG] - Module::Answer2Pix - 实现 `answer2pix` 函数，将 `LaTeX` 公式转换为对应的图像。
- `view.py` :
 - 2024-05-24 08:45:22.345 TRACE [MainThread] [DEBUG] - View::Initialization - 定义 `LatexViewer` 类，创建 GUI 界面组件。
 - 2024-05-24 09:05:34.678 TRACE [MainThread] [DEBUG] - View::ImageUpload - 实现图像上传、`LaTeX` 公式显示和计算结果显示的功能。
 - 2024-05-24 09:25:44.123 TRACE [MainThread] [DEBUG] - View::RestartApp - 提供重新启动应用程序的功能。
- `test.py` :
 - 2024-05-24 10:15:56.789 TRACE [MainThread] [DEBUG] - Test::Functionality - 编写测试代码，验证图像识别和公式计算功能。

4.5 未来计划

1. 增加更多数学公式的支持和复杂计算功能。
2. 优化图像处理速度和识别准确率。
3. 提升 GUI 界面的用户体验，增加更多交互功能。

5 项目收获与体会

1. 井旺波

- 经过本学年科学与社会研讨课的学习，我们组进行的“数学公式的识别与计算”项目顺利完成，在和同组队员的共同努力下，我收获了许多，下面是我的心得体会，以及自身收获。

首先，在项目开始的初期，我通过B站，慕课等平台对python进行了较为系统的学习，初步了解了python相关知识并且学会利用python软件如PyCharm、VScode等软件进行初步的python编程。此外，我还在同学的帮助下学会了GitHub等程序开源网站的使用方法，收获颇丰。

其次，项目中我分配到了图形化界面设计与制作的工作，因此又进一步在网络上自主学习了tkinter的使用方法，初步掌握了利用python语言设计制作图形化界面的能力。另外，在制作图形化界面的过程中，我还学会了使用通义灵码等智能AI辅助编程。通过AI纠错，智能生成代码等功能，大大减轻了工作量，同时提高了工作质量。

最后，我非常感谢在项目进行过程中同学们对我的帮助，我之所以能够成功从对编程一窍不通到如今能够编写一定的程序，除了个人在线下以及线上课程的学习外，也离不开研讨课同学们的帮助。而且在本学年与同学们的深度合作中，我深刻体会到了团队中每个人各司其职，分工协作的重要意义，正是因为本学年大家通力合作，才使得项目能够圆满完成。

2. 邱天爽

- 在本次项目中，我负责了代码的最终整合与调试工作，同时承担了PPT和报告的制作任务，并参与了部分后端代码的编写。这次项目经验让我受益匪浅。

首先，通过与小组成员的合作与分工，我学会了如何系统地整合一个项目，如何合理分配工作，以及如何平衡各个任务，使每个成员都能充分发挥自己的特长。在能力提升方面，我的Python编程能力得到了显著提高，同时也强化了我的软件工程管理方法。调试过程中，我的查错能力也得到了有效锻炼，为以后的学习和研究打下了坚实的基础。

此外，这次项目让我完整地体验了从无到有的开发过程，学习和使用了前人的科研成果，并在此过程中掌握了许多新知识。这些宝贵的经验将对我未来的学习和工作产生深远的影响。

3. 魏一凡

- 在本次项目开发经历中，我学习到最多的是关于整个项目统筹安排的经验。知道了一个项目的整体开发模式、项目架构实际上是整个项目最重要的内容之一，一个没有规划好的项目，在开发过程中总会遇到这样的问题；只有明晰了每个人的任务，以及整个项目的每个流程，项目才能顺利通关。

此外，我深刻地认识到了AI对于程序员工作的影响之深，虽然最终代码的每一行我都进行了细致的审查，但其中最枯燥的编写代码部分仍是大量借助了AI代码助手的力量。可以说，有了AI的帮助，我编写代码的效率至少提升了80%。

对我而言，这段经历是一次全方位的提升。在与团队的紧密协作中，我学会了如何更好地沟通与协调，如何在复杂的问题面前寻找最优解。我投入了大量精力于项目细节，从构思到实施，每一步都力求精益求精，我相信这些努力在未来的学习中都能得到回报。

4. 张显

- 在本次科学与社会研讨课中，我们小组设计开发了一款具有图形识别和公式计算功能的数学软件。在学习和开发过程中，我们收获颇丰。

数学公式的计算在学术研究、教育和工程实践中占有重要地位，然而手动输入公式不仅费时费力而且容易出错。为了降低数学公式计算的成本，我们决定开发本软件。

在项目开发的最开始，我们讨论了项目应有的各个功能，并根据各个功能的设计进行了小组分工。我所负责的部分为识别结果转化与结果计算。

在开发过程中，我学会了上网查找、学习和使用python库，这大大加快了我们的开发进度。利用他人已经开发完善的python库使得我们能够在较短的时间内完成项目主体的代码设计。

同时，由于不同同学负责部分不同，我们在开发过程中遇到了沟通上的问题，这是之前单独写程序时没有遇到过的。在与其他同学的交流沟通过程中，我对较大程序的开发过程有了新的认知，对合作开发项目有了新的了解。

通过本次社会与科学研讨课，我对如何利用自己的知识解决科学问题有了更多的认识，对合作完成任务有了更多认识。

5. 周根发

- 经过近一年的选题、学习与实践，我们最终完成了Pix2Answer项目，也给我们“科学与社会”研讨课画下了一个完美的句号。在这一年中我从对计算机编程的一无所知到初步了解，再到能与他人合作完成小组项目，获得了极大的进步。首先便是初步学习了Python语言，学会了基本的语法，会调用一些简单的库函数，同时还学会了使用Github等帮助编程的平台和软件。其次，我们学会了分工合作，每个人负责不同的部分，学会了如何高效的与人沟通。最后，我们也学会了如何合理安排任务进度与时间管理。总之，这次的项目对我而言是一次挑战也是机遇，为我打开了一扇通往计算机世界的大门。

6 结论与展望

通过本项目，我们成功开发了 Pix2Answer，一个能够自动识别和处理数学公式的图像处理系统。该系统在识别准确性、处理速度和用户体验方面都达到了预期目标。然而，仍有一些改进空间，例如对复杂公式的识别准确性和对更多图像格式的支持。未来，我们计划进一步优化识别算法，并扩展系统的功能，使其在更多应用场景中发挥作用。

7 致谢

感谢中国科学技术大学的支持与指导，特别是我们的指导老师陈欢欢教授。在项目的开发过程中，团队成员齐心协力，克服了重重困难，共同完成了 Pix2Answer 系统的设计与实现。希望此项目能够在未来的发展中，持续优化，解决更多实际问题。

参考文献：

- [1] Matplotlib 编者. Matplotlib: Visualization with Python [G/OL]. Matplotlib Documentation, 2023 [2024-07-09]. <https://matplotlib.org/stable/contents.html>.
- [2] Pillow 编者. Pillow (PIL Fork) Documentation [G/OL]. Read the Docs, 2023 [2024-07-09]. <https://pillow.readthedocs.io/en/stable/>.
- [3] Blecher, Lukas 编者. pix2tex: Latex OCR [G/OL]. GitHub Repository, 2023 [2024-07-09]. <https://github.com/lukas-blecher/LaTeX-OCR>.
- [4] Toman, August 编者. latex2sympy2 [G/OL]. GitHub Repository, 2023 [2024-07-09]. <https://github.com/augustt198/latex2sympy>.
- [5] Python Software Foundation 编者. Tkinter – Python interface to Tcl/Tk [G/OL]. Python Documentation, 2023 [2024-

07-09]. <https://docs.python.org/3/library/tkinter.html>.

[6] SymPy Development Team 编者. SymPy: Python Library for Symbolic Mathematics [G/OL]. SymPy Documentation, 2023 [2024-07-09]. <https://docs.sympy.org/latest/index.html>.

8 附录

8.1 附录A：项目文档

8.1.1 README.md

```
1 # Pix2Answer: 图像中的数学公式识别与计算工具
2 ## 项目简介
3 Pix2Answer 是一个创新的数学辅助工具，旨在帮助学生和教育工作者快速识别并计算图片中的
  数学公式。通过结合先进的图像处理技术和数学解析算法，该工具能够将照片中的数学表达
  式转化为可计算的LaTeX格式，进而给出计算结果。项目由 JingWangBo、TorryQ、
  YorkyifanWei、ZhouGenFa 和 ZhangXian 联合开发，旨在简化数学问题的解答过程，提升
  学习效率。
4 ## 功能亮点
5 - 公式识别：利用 pix2tex 库的 OCR 技术，精准识别图片中的数学公式。
6 - 公式转换与计算：通过 latex2sympy2 和 sympy，将识别的公式转换为可计算的数学表达
  式，并执行计算。
7 - 结果可视化：使用 matplotlib 生成计算结果的图像表示，直观展示。
8 - 用户界面：构建了基于 tkinter 的图形界面，便于用户上传图片、查看识别的公式及计算
  结果。
9 ## 技术栈
10 - 图像处理：Pillow (PIL) 用于图像的读取与基本处理。
11 - 公式识别：pix2tex 的 LatexOCR 类实现图像到LaTeX的转换。
12 - 数学计算：latex2sympy2 与 sympy 处理数学公式解析与计算。
13 - GUI开发：tkinter 构建跨平台的图形用户界面。
14 ## 项目结构
15 - main.py: 项目主入口，负责初始化GUI界面及图像处理逻辑。
16 - module.py: 核心功能模块，包含公式识别与图像生成函数。
17 - view.py: 定义GUI视图，实现用户交互组件。
18 - test.py: 测试脚本，验证核心功能的正确性。
19 - .gitignore: 配置文件，排除无关文件于版本控制之外。
20 - README.md: 当前文档，项目说明与指南。
21 ## 使用教程
22 1. 安装依赖：确保安装所有必要的Python库，通过运行`pip install pillow
  matplotlib sympy pix2tex`完成。
23 2. 运行程序：在命令行中，切换到项目根目录，执行 python main.py 启动应用。
24 3. 上传图片：通过GUI界面的“上传图片”按钮，选择含有数学公式的图片。
25 4. 查看结果：应用将显示原始公式、计算结果及结果图片（如果适用）。
26 ## 依赖版本
27 - pix2tex      -- v0.1.2
28 - latex2sympy2 -- v1.9.1
29 - matplotlib   -- v3.9.0
30 - tkinter
31 ## 未来展望
32 - 扩展支持更多数学符号与复杂公式的识别。
33 - 优化图像处理算法，提升识别速度与准确率。
34 - 引入更丰富的用户交互设计，提升用户体验。
35 ## 致谢
```

36 | 感谢团队成员的共同努力，老师持续的支持与指导，以及开源社区提供的强大工具和库，使得该项目得以实现。

8.2 附录B：项目文件结构

```
1 | .pix2Answer/  
2 | └─ .gitignore  
3 | └─ README.md  
4 | └─ main.py  
5 | └─ module.py  
6 | └─ test.py  
7 | └─ view.py
```

8.3 附录C：项目主要代码

8.3.1 main.py

```
1 | # 导入view模块，用于界面显示  
2 | import view  
3 | # 从PIL导入Image类，用于图像处理  
4 | from PIL import Image  
5 | # 导入tkinter模块，用于创建GUI应用程序  
6 | import tkinter as tk  
7 | # 从pix2tex的CLI导入LatexOCR类，用于OCR识别数学公式  
8 | from pix2tex.cli import LatexOCR  
9 | # 导入module模块，用于公式转换和图像生成  
10 | import module  
11 |  
12 |  
13 | def process_image(image_path, callback):  
14 |     """  
15 |     处理图像，将其转换为LaTeX代码并进行进一步处理。  
16 |  
17 |     参数：  
18 |     image_path (str): 图像文件的路径。  
19 |     callback (function): 处理完成后的回调函数，接收LaTeX输入、输出及生成的图像作为参数。  
20 |  
21 |     返回：  
22 |     无  
23 |     """  
24 |     # 打印处理开始信息  
25 |  
26 |     # # 示例处理逻辑  
27 |     # print("Processing image:", image_path)  
28 |  
29 |     # 打开图像文件  
30 |     img = Image.open(image_path)  
31 |     # 初始化OCR对象，用于识别图像中的LaTeX代码  
32 |     latex_ocr = LatexOCR()  
33 |     # 使用OCR识别图像中的LaTeX代码  
34 |     latex_input = latex_ocr(img)  
35 |  
36 |     # # 打印识别得到的LaTeX输入  
37 |     # print("Latex input:", latex_input)  
38 |  
39 |     # 使用module模块中的函数处理LaTeX输入，得到LaTeX输出
```

```

40     latex_output = module.pix2answer(latex_input)
41
42     # # 打印处理后的LaTeX输出
43     # print("Latex output:", latex_output)
44
45     # 根据LaTeX输出生成对应的图像
46     external_image = module.answer2pix(latex_output)
47     # 调用回调函数, 传入LaTeX输入、输出及生成的图像
48     # 通过回调函数通知view.py处理完成
49     callback(latex_input, latex_output, external_image)
50
51 # 示例调用, 实际中应由view.py调用
52 # process_image("path/to/image.jpg", update_view_callback)
53
54
55 def main():
56     """
57     主函数, 创建GUI应用程序的入口。
58     """
59     # 创建根窗口
60     root = tk.Tk()
61     # 初始化界面视图
62     app = view.LatexViewer(root)
63     # 运行主事件循环
64     root.mainloop()
65
66
67 if __name__ == "__main__":
68     main()

```

8.3.2 view.py

```

1  import tkinter as tk
2  from tkinter import filedialog
3  from PIL import Image, ImageTk
4  import main
5
6
7  class LatexViewer:
8      """
9      LaTeX查看器类, 用于显示和处理 LaTeX 公式。
10
11      Attributes:
12          master: tkinter 主窗口对象。
13          image_path: 选择的图像文件路径。
14          latex_input: 输入的 LaTeX 公式。
15          latex_output: 处理后的 LaTeX 公式。
16          external_image: 处理后的外部图像。
17          process_image_callback: 图像处理回调函数。
18      """
19      def __init__(self, master):
20          """
21          初始化 LaTeX 查看器。
22
23          参数:
24              master: tkinter 主窗口对象。

```



```

25         """
26         self.restart_button = None
27         self.show_image_button = None
28         self.display_result_button = None
29         self.display_formula_button = None
30         self.choose_image_button = None
31         self.master = master
32         self.master.title("Pix2Answer")
33         self.master.geometry("600x800")
34         self.create_widgets()
35         self.image_path = ""
36         self.latex_input = ""
37         self.latex_output = ""
38         self.external_image = None
39         self.process_image_callback = None
40
41     def create_widgets(self):
42         """
43         创建界面组件。
44         """
45         self.choose_image_button = tk.Button(self.master, text="上传图片",
command=self.open_file_dialog)
46         self.choose_image_button.pack()
47
48         self.display_formula_button = tk.Button(self.master, text="显示原始公式",
49 state=tk.DISABLED)
command=self.show_latex_input,
50         self.display_formula_button.pack(side="left", expand=False, padx=3, pady=1)
51
52         self.display_result_button = tk.Button(self.master, text="显示计算结果",
53 state=tk.DISABLED)
command=self.show_latex_output,
54         self.display_result_button.pack(side="right", expand=False, padx=3, pady=1)
55
56         self.show_image_button = tk.Button(self.master, text="显示结果图片",
57 state=tk.DISABLED)
command=self.show_external_image,
58         self.show_image_button.pack(side="right", expand=False, padx=3, pady=1)
59
60         self.restart_button = tk.Button(self.master, text="Return to start",
61 command=self.restart_app)
62         self.restart_button.pack(side="bottom", padx=10, pady=10, anchor="se")
63
64     def open_file_dialog(self):
65         """
66         打开文件选择对话框，并显示所选图像。
67         """
68         self.image_path = filedialog.askopenfilename()
69         if self.image_path:
70             image = Image.open(self.image_path)
71             image.thumbnail((300, 300)) # 缩放图片
72             photo = ImageTk.PhotoImage(image)
73             label = tk.Label(self.master, image=photo)
74             label.image = photo
75             label.pack(pady=2)

```

```

76         self.process_image_callback = lambda latex_in, latex_out, external_image:
(
77             self.on_image_processed(latex_in, latex_out, external_image))
78         main.process_image(self.image_path, self.process_image_callback)
79
80     def on_image_processed(self, latex_input, latex_output, external_image):
81         """
82         处理图像完成后调用的函数，用于更新界面状态。
83
84         参数：
85             latex_input: 输入的 LaTeX 公式。
86             latex_output: 处理后的 LaTeX 公式。
87             external_image: 处理后的外部图像。
88         """
89         self.latex_input = latex_input
90         self.latex_output = latex_output
91         self.external_image = external_image
92         self.update_ui_state(True)
93
94     def update_ui_state(self, enable=True):
95         """
96         更新界面组件的状态（启用或禁用）。
97
98         参数：
99             enable: 是否启用组件，默认为 True。
100         """
101         state = tk.NORMAL if enable else tk.DISABLED
102         self.display_formula_button.config(state=state)
103         self.display_result_button.config(state=state)
104         self.show_image_button.config(state=state)
105
106     def show_latex_input(self):
107         """
108         显示输入的 LaTeX 公式。
109         """
110         if self.latex_input:
111             label = tk.Label(self.master, text=self.latex_input)
112             label.pack()
113
114     def show_latex_output(self):
115         """
116         显示处理后的 LaTeX 公式。
117         """
118         if self.latex_output:
119             label = tk.Label(self.master, text=self.latex_output)
120             label.pack()
121
122     def show_external_image(self):
123         """
124         显示处理后的外部图像。
125         """
126         if self.external_image:
127             label = tk.Label(self.master, image=self.external_image)
128             label.image = self.external_image
129             label.pack()
130

```

```

131     def restart_app(self):
132         """
133         重启应用程序。
134         """
135         self.master.destroy()
136         import main
137         main.main()
138
139
140 if __name__ == "__main__":
141     root = tk.Tk()
142     app = LatexViewer(root)
143     root.mainloop()

```

8.3.3 module.py

```

1  # 导入必要的库，用于图像处理和数学公式转换
2  import io
3  import os
4  from tempfile import NamedTemporaryFile
5
6  from latex2sympy2 import latex2latex
7  from matplotlib import pyplot as plt
8  from matplotlib.figure import Figure
9  from PIL import Image, ImageTk
10
11
12 def pix2answer(pix):
13     """
14     将像素值转换为LaTeX公式。
15
16     参数:
17     pix -- 字符串形式的像素值
18
19     返回:
20     answer -- 经过处理的LaTeX公式字符串
21     """
22     answer = latex2latex(pix)
23     return answer
24
25
26 def answer2pix(answer):
27     """
28     将数学公式转换为图像像素。
29
30     参数:
31     answer -- LaTeX公式字符串
32
33     返回:
34     tk_image -- Tkinter兼容的图像对象
35     """
36     # print(answer)
37
38     # 创建一个不带边框的空白图形
39     fig = Figure()
40     ax = fig.add_axes((0, 0, 1, 1), frameon=False, aspect=1)

```

```
41     # 设置文本属性
42     fontsize = 16
43     color = "black"
44     ax.text(0.5, 0.5, answer, fontsize=fontsize, color=color, ha="center",
45     va="center",
46         transform=ax.transAxes, usetex=True)
47
48     # 使用临时文件保存图像
49     # 使用with语句确保文件自动关闭
50     with NamedTemporaryFile(dir=os.path.join(os.getcwd()), suffix=".png",
51     delete=False) as temp_file:
52         # 更新图形并保存为PNG图像
53         plt.draw()
54         plt.savefig(temp_file.name, format="png", dpi=300, bbox_inches="tight")
55
56         # 打开图像文件并转换为Tkinter兼容的格式
57         # 打开图像并立即关闭
58         image = Image.open(temp_file.name)
59         image.load() # 确保图像数据完全加载
60         # 立即转换为Tkinter兼容的图像格式，然后关闭原始PIL图像
61         tk_image = ImageTk.PhotoImage(image=image)
62         image.close() # 显式关闭图像
63         # print("Image saved to", temp_file.name)
64
65     # 尝试删除文件，此时临时文件应该已经关闭
66     try:
67         os.remove(temp_file.name)
68     except OSError as e:
69         print(f"Error: {e.strerror}. Unable to delete temporary file:
70     {temp_file.name}")
71
72     return tk_image
```