gcodepainter

🌟 Introduction and Motivation

I am Matt Curtis, known as PhysCorp on GitHub. I developed gcodepainter, a tool designed to convert PNG images into GCODE for plotting machines.

🔍 Overview of gcodepainter

gcodepainter transforms PNG images into GCODE, suitable for plotting machines. It's useful for those interested in 3D printing and plotting.

💻 Environment and Dependencies

- **Managed via Conda**
- **Python Version:** 3.8
- **Key Libraries:** opencv-python, numpy, flask, rich
- **Installation:** See `README` in the repository.

🛠️ Project Usage

- **Basic Operation:** Simple commands, Flask-based GUI.
- **Pronterface Integration:** For Windows only.
- **Configurable Settings:** Detailed in `README`.

🧪 Methods and Implementation

Practical Application Focus

Designed for ease of use in GCODE generation and image processing.

🏗️ Technical Structure

- **WebUI Class:** Manages website using Flask.
- **AutoClass:** Image processing with opencv.
- **Third-Party Libraries:** Flask, opencv, numpy.
- **File I/O Operations:** GCODE saving and error handling.

🖼️ Visualization and Interaction

The Flask web interface visualizes image processing steps and allows users to upload images, view outputs, and download/print GCODE.

🎉 Project Achievements

Fulfillment of Requirements (outlined verbally & in report)

- Practical Application
- Development of Functions
- Third-party Library Usage
- File I/O with Error Handling
- Visualization of Outcomes

✨ Additional Features

- **Pronterface Support:** For Windows.
- **Community Engagement:** Open-source on GitHub.

📈 Results and Impact

- **User Interface:** Command-line tool and web interface.
- **Process Visualization**
- **GCODE Interaction**
- **Community Engagement**

🏁 Conclusion

gcodepainter serves as a functional and educational tool, encouraging community involvement in 3D printing, plotting, and image processing.