ComfyUI Workflow Embedder (workflow_embedder.py)

This Python script is a utility for image generation enthusiasts using ComfyUI. It performs the reverse of standard workflow extraction: it takes an existing base PNG image and a separate JSON file containing a ComfyUI workflow, embeds the JSON into the image's metadata, and saves the result as a new PNG.

This allows you to create high-quality, shareable images that contain all the necessary generation "recipe" data for others to reproduce your results.

🚀Key Features

- **Embed Workflow:** Takes a standard ComfyUI JSON file and writes its content into the image's PNG metadata chunk (comfyui_workflow key).
- **Preserve Data:** It safely preserves any existing metadata in the input image while adding the new workflow data.
- **Command Line Interface:** Uses argparse for clean, reliable command-line execution with mandatory input/output paths.

🛠️Requirements

The script relies on the Pillow (PIL fork) library for image and metadata manipulation.

Prerequisites

1. **Python:** Python 3.x must be installed.
2. **Pillow:** You must install the Python Imaging Library (Pillow).

To install Pillow, run:

```
pip install Pillow
```

💡Usage

The script requires three mandatory positional arguments: the input image, the input JSON workflow, and the desired output image file.

Command Structure

```
python workflow_embedder.py <INPUT_IMAGE_PATH> <INPUT_JSON_PATH>
<OUTPUT_IMAGE_PATH>
```

Example

Using the file names from our development session:

```
python workflow_embedder.py ComfyUI_00231_bear_snow.png flux-schnell-fast-workflow-cabin-SNOW.json ComfyUI_00231_bear_snow_e.png
```

Argument	**Description**	**Example Value**
INPUT_IMAGE_PATH	The base PNG image to embed the workflow into.	ComfyUI_00231_bear_snow.png
INPUT_JSON_PATH	The ComfyUI workflow saved as a JSON file.	flux-schnell-fast-workflow-cabin-SNOW.json
OUTPUT_IMAGE_PATH	The name of the new PNG file that will contain the embedded workflow.	ComfyUI_00231_bear_snow_e.png

🔍Implementation Details (For Developers)

The script uses a critical component of the Pillow library to ensure metadata compatibility:

The PIL.PngImagePlugin.PngInfo Fix

Standard Python dictionaries cannot be passed directly to the PNG saving function for metadata, which often results in the error: 'dict' object has no attribute 'chunks'.
To solve this, the script correctly uses the PngInfo class:

1. from PIL.PngImagePlugin import PngInfo is imported.
2. A PngInfo() object is initialized.
3. Existing metadata (from img.info) and the new workflow string are added using pnginfo.add_text(key, value).
4. The img.save() call uses the pnginfo=pnginfo keyword argument, ensuring the metadata is correctly formatted as PNG "chunks."