Vi3W

Task-1

Documentation: Steps, Challenges, and Optimizations in Setting up Stable Fast 3D

i) Setting Up the Environment

1. Create a Virtual Environment

- Created an isolated environment using: **python -m venv sf3d_env**.
- Activated the environment:
  - **Windows: sf3d_env\\Scripts\\activate**

2. Update PATH

- Checked the Python path with **where python** to make sure the interpreter in the right order, which is **sf3d_env**.
- Successfully updated the value of PATH to point to the virtual environment's directory, **Scripts**.

---

ii) Installing Dependencies

1. Requirements Installation

- Ran **pip install -r requirements.txt** for installation of dependencies.
- Modified the **requirements.txt** to exclude GPU dependencies:
  - Removed **rembg[gpu]** and instead added: **rembg==2.0.57; sys_platform == 'darwin'.**

2. Install PyTorch Compatible with CPU

- PyTorch, TorchVision, and Torchaudio were installed using the following command:
  - **pip install torch==2.5.1+cpu torchvision==0.20.1+cpu torchaudio==2.5.1+cpu --index-url https://download.pytorch.org/whl/cpu**

---

iii) Troubleshooting uv_unwrapper

1. **Error Encountered**
   - The script reported:
     - ImportError: cannot import name 'Unwrapper' from 'uv_unwrapper'.
2. **Steps to Resolve**
   - Verified that setup.py exists in the directory uv_unwrapper.
     - Installed uv_unwrapper by:
     - pip install ./uv_unwrapper/
   - If torch was missing after installation, reinstalled the torch using the CPU version.
3. **Alternative Resolution**
   - Added the following to the Python path: **uv_unwrapper** directory:
     - **import sys**
     - **sys.path.append('F:/ComfyUI-master/custom_nodes/stable-fast-3d/uv_unwrapper')**

---

iv) Hugging Face Authentication

1. **Create Token**
   - Created a token off of the Hugging Face Token Page.
2. **Log in**
   - Authenticated with huggingface-cli login.
   - Set up **HF_TOKEN** as an environment variable:
     - **set HF_TOKEN=<your_token>**

---

v) Running the Script

1. Execution
   - Ran the main script with:
     - **python run.py demo_files/examples/chair1.png --output-dir output/ --device cpu**
2. **Testing Output**
   - Verified that the generated **.glb** file imports correctly in Blender.

vi) Challenges Faced and Resolutions

1. **Torch Import Issues**
    - **Problem**: **ModuleNotFoundError: No module named 'torch'** even after installation.
    - **Resolution**:
        - Verified the virtual environment activation.
        - Reinstalled **torch** for the correct platform, CPU.
2. **Conflicting Python Paths**
    - **Issue**: Additional Python installs disrupted module resolution.
    - **Resolution**:
        - Updated PATH to prioritize the virtual environment.
3. **Hugging Face Token**
    - **Issue**: **Invalid token** error during authentication.
    - **Resolution**:
        - Pasted the token correctly and logged in using the CLI.
        - Made sure tokens and permissions are valid on Hugging Face.
4. **Dependency Installation Issues**
    - **Issue**: **subprocess-exited-with-error** while attempting installation of **uv_unwrapper**.
    - **Resolution**:
        - Manually installed the required dependencies first before trying to install **uv_unwrapper**.

---

vii) Key Learnings and Optimizations

1. **Effective Dependency Management**
    - Using CPU-specific dependencies for a system that has no GPU avoids resource constraints.
2. **Isolation in Environment**
    - Ensure a clean and isolated environment to prevent conflicts.
3. **Troubleshooting Python Imports**
    - Tools like **pip list**, **where python**, and Python's **sys.path** are very useful during debugging.

4. **Authentication and Access**
    - Setting environment variables for tokens provides a more seamless workflow.

Task-2

Documentation: Creating a One-Click Installer

Overview

This documentation outlines the steps for developing a one-click installer for the **Stable Fast 3D** project. The installer automates the setup process, including dependency installation, environment configuration, and ensuring offline functionality. The goal is to minimize user input and make the project deployment seamless on any machine.

Steps to Develop a One-Click Installer

Step 1: Understand the Environment

- **Project Dependencies**:
    - Python 3.12 or later.
    - Required libraries (e.g., torch, torchvision, rembg, etc.).
    - Local installations like uv_unwrapper and texture_baker.
- **Challenges Identified**:
    - Compatibility with CPU-only environments.
    - Offline installation for dependencies like uv_unwrapper and texture_baker.
    - Avoiding issues with environment variables and Python paths.

Step 2: Prepare Resources

- Collect and bundle all required dependencies in a requirements.txt file or pre-download wheels for offline installation.
- Gather the necessary Python scripts and ensure they are executable in any environment.

Step 3: Write the Installer Script

Installer Script: Key Functionalities

1. **Create and Activate a Virtual Environment**:

```
python -m venv sf3d_env
source sf3d_env/bin/activate  # Linux/Mac
sf3d_env\Scripts\activate     # Windows
```

2. **Install Dependencies**:
    - Online mode:

pip install -r requirements.txt

- Offline mode: Use pre-downloaded wheels:

pip install ./dependencies/*.whl

3. **Install Local Modules**:

```
pip install ./uv_unwrapper/
pip install ./texture_baker/
```

4. **Set Environment Variables (if needed)**:
   - Update PATH to include the Python environment:

   setx PATH "%PATH%;F:\ComfyUI-master\custom_nodes\stable-fast-3d\sf3d_env\Scripts"

5. **Verify Installation**: Run a test command:

python run.py demo_files/examples/chair1.png --output-dir output/ --device cpu

---

Script Implementation

Here's the **installer script**:

```
import os
import subprocess
import sys

def run_command(command):
    """Helper function to execute system commands."""
    try:
        subprocess.run(command, shell=True, check=True)
    except subprocess.CalledProcessError as e:
        print(f"Error: {e}")
        sys.exit(1)

def main():
    print("Starting installation...")

    # Step 1: Create virtual environment
    if not os.path.exists("sf3d_env"):
        print("Creating virtual environment...")
        run_command("python -m venv sf3d_env")
    else:
        print("Virtual environment already exists.")

    # Step 2: Activate the environment
    activate_command = (
```

```
        ".\\sf3d_env\\Scripts\\activate" if os.name == "nt" else "source sf3d_env/bin/activate"
    )
    print("Activating virtual environment...")
    run_command(activate_command)

    # Step 3: Install dependencies
    print("Installing dependencies...")
    run_command("pip install -r requirements.txt")

    # Step 4: Install local modules
    print("Installing local modules...")
    run_command("pip install ./uv_unwrapper/")
    run_command("pip install ./texture_baker/")

    # Step 5: Verify installation
    print("Verifying installation...")
    run_command("python run.py demo_files/examples/chair1.png --output-dir output/ --
device cpu")

    print("Installation complete!")

if __name__ == "__main__":
    main()
```

Challenges and Solutions

- **Issue**: Missing modules like torch during local module installations.
  - **Solution**: Ensure **torch** and dependencies are explicitly installed before running the installer.
- **Issue**: Dependency conflicts or version mismatches.
  - **Solution**: Pin versions in **requirements.txt** and validate compatibility offline.
- **Issue**: Offline functionality.
  - **Solution**: Bundle all dependencies as **.whl** files in a **dependencies/** directory.

Testing the Installer

1. **Run the script**:

   ```
   python installer.py
   ```

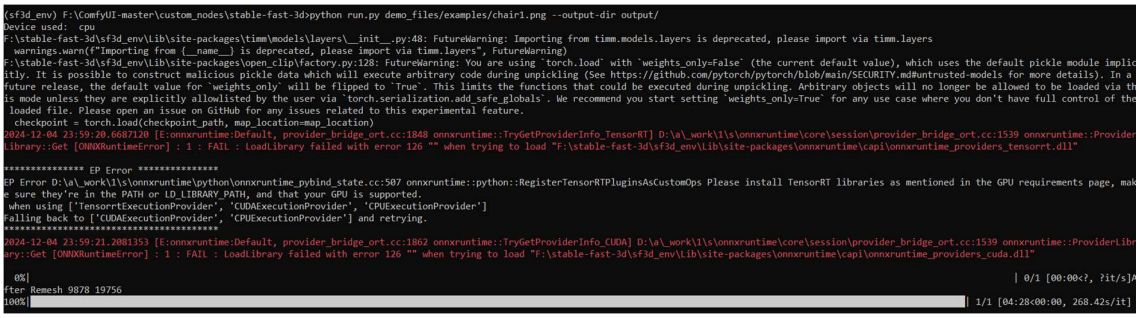
2. **Verify output**:

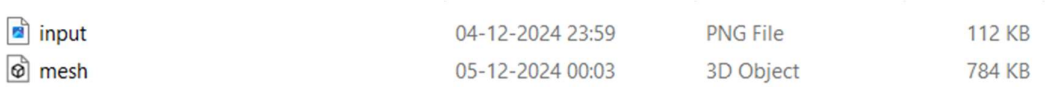
   - Ensure the virtual environment is created.
   - Check that dependencies are installed without errors.
   - Confirm the test command runs successfully.

Improvements and Optimizations

- **Bundle Resources**:
  - Package the installer and dependencies into a .zip file for easy distribution.
- **Error Handling**:
  - Add more detailed logs for troubleshooting.
- **Cross-Platform Support**:
  - Test the script on Windows, Linux, and Mac environments.

Screenshots:





Name: Pulkit Agrawal

Contact: +91 7340025509

E-mail: pulkitag2003@gmail.com