根据目前最新的代码，写个详细的软件使用说明书

回答

好的，用户让我根据最新的代码写一份详细的软件使用说明书。首先，我需要回顾之前的对话和代码，确保我完全理解软件的功能和使用方法。用户提供的代码是一个用于下载 Docker 镜像的 Python 脚本，支持多线程、自动处理认证错误、分块下载、校验和验证等功能。

首先，我需要明确用户的需求。他们需要一个结构清晰、内容详尽的说明书，可能包括简介、功能列表、环境要求、安装步骤、使用说明、参数详解、错误处理、常见问题等部分。用户可能是开发者或运维人员，需要快速上手使用这个工具，所以说明书需要简明扼要，同时覆盖所有关键点。

接下来，我要整理软件的主要功能点。根据代码，主要功能包括多线程下载、自动处理 HTTP 416 错误、连接池管理、镜像解析、认证处理、进度显示、校验和验证等。这些都需要在说明书中详细描述，特别是用户可能关心的特性，比如多线程加速和错误恢复机制。

然后，考虑用户可能的使用场景。例如，用户可能需要从私有仓库下载镜像，或者处理网络不稳定的情况。说明书需要指导用户如何设置仓库地址、调整线程数，以及如何处理下载中的各种错误。同时，用户可能需要了解如何验证下载的文件完整性，这也是需要强调的部分。

环境要求方面，需要明确 Python 版本（3.8+）和必要的依赖库，如 requests、urllib3、tqdm 等。用户可能不清楚如何安装这些依赖，所以需要给出具体的 pip 安装命令。

安装步骤要简单明了，包括克隆仓库、安装依赖、赋予执行权限（如果有的话）等。对于不同操作系统的用户，可能需要不同的说明，但用户提供的代码是 Python 脚本，跨平台通用，所以可能不需要特别区分。

参数详解部分需要详细说明每个命令行选项的作用，如镜像名称格式、仓库地址、线程数限制等。用户可能在输入镜像名称时遇到格式问题，需要给出示例，比如“infiniflow/ragflow:v0.17.0”或“nginx:latest”。

错误处理和常见问题部分要预见用户可能遇到的问题，比如认证失败、网络超时、文件校验失败等，并提供解决方案。例如，遇到 416 错误时，工具会自动重试，用户不需要

> 手动干预，但可能需要了解背后的机制。
>
> 此外，用户可能对日志文件的位置和内容感兴趣，需要说明日志记录的位置和如何查看日志信息。
>
> 最后，需要确保说明书的格式美观，结构清晰，使用标题、列表、代码块等元素，提高可读性。同时，语言要简洁，避免技术术语过多，让不同层次的用户都能理解。
>
> 总结来说，我需要按照标准的软件说明书结构组织内容，涵盖简介、功能、安装、使用、参数、示例、错误处理、常见问题等部分，确保用户能够顺利使用该工具，并解决可能遇到的问题。

Docker 镜像下载工具使用说明书

一、工具简介

本工具是针对 Docker 镜像仓库设计的自动化下载工具，支持从公共/私有仓库高效下载镜像层文件。主要特性包括：

- 多线程加速下载
- 自动处理 **HTTP 416** 错误
- 智能连接池管理
- 镜像名称自动解析
- 仓库认证自动处理
- 实时下载进度显示
- 文件完整性校验
- 错误自动重试机制

二、环境要求

- **Python 3.8+**

- 依赖库：requests, urllib3, tqdm

安装依赖
pip install requests urllib3 tqdm

三、安装步骤

1.下载最新代码文件 docker_image_puller.py

2.安装依赖包

3.赋予执行权限（可选）

chmod +x docker_image_puller.py

四、使用说明

1. 基础命令格式

python docker_image_puller.py <镜像名称> [选项]

2. 参数详解

参数	必填	说明	示例
镜像名称	是	支持标准镜像格式	infiniflow/ragflow:v0.17.0nginx:latest
-r/--registry	否	仓库地址（默认 docker 官方仓库）	docker.1ms.run
-t/--threads	否	下载线程数（1-20，默认 5）	10

3. 使用示例

```
# 从默认仓库下载 nginx 最新版
python docker_image_puller.py nginx:latest

# 从私有仓库下载镜像（10 线程）
python docker_image_puller.py infiniflow/ragflow:v0.17.0 -r docker.1ms.run -t 10
```

五、核心功能说明

1. 下载过程可视化

- 实时显示：
- 每个层的 12 位摘要标识
- 下载百分比进度
- 已用/剩余时间估算
- 全局进度：

```
[Global] 3/5 layers | Elapsed: 00:02:17
```

2. 文件存储结构

下载文件按以下结构组织：

```
[镜像名称]_layers/
└── [摘要前 12 位]/
    └── layer.tar
```

3. 错误处理机制

错误类型	处理策略	重试间隔
HTTP 416	自动删除损坏文件刷新连接池重新认证	指数退避
网络超时	自动重试	5 秒~10 分钟
校验失败	终止下载	-

六、高级配置

修改代码顶部参数调整性能：

```
# 连接配置
CONNECT_TIMEOUT = 600    # 连接超时(秒)
READ_TIMEOUT = 7200      # 读取超时(秒)

# 重试策略
MAX_RETRIES = 20         # 最大重试次数
RETRY_BACKOFF_BASE = 5   # 退避基数

# 性能调优
INIT_CHUNK_SIZE = 262144 # 256KB 初始块
LARGE_FILE_THRESHOLD = 524288000  # 500MB 大文件阈值
```

七、日志监控

- 日志文件：自动生成 docker_puller.log

- 关键日志标记：

```
[WARNING] 第 3 次重试 a3ed95caeb02 原因: 504 Gateway Timeout 等待: 18.7s
[ERROR] 校验失败: 期望 sha256:a3ed95ca 实际 sha256:532ded5d
```

八、常见问题解答

Q1: 出现 SSL 证书验证失败错误

```
# 临时解决方案（测试环境）
export PYTHONWARNINGS="ignore:Unverified HTTPS request"
```

Q2: 下载进度卡在某个百分比

1.检查网络连接

2.查看日志文件中的错误提示

3.尝试减少线程数 -t 3

Q3: 如何验证下载的完整性？

所有层下载完成后会自动进行 SHA256 校验，可通过日志查看：

[INFO] sha256:a3ed95ca 下载完成

九、技术支持

遇到未解决问题时请提供：

1.完整的命令行参数

2.docker_puller.log 文件内容

3.出现问题的镜像名称

工具版本：v3.7.2

最后更新：2025-03-04