**Wie eine lokale KI die Fotosammlung auf dem NAS verschlagworten kann**

Fotos sind mit dem Smartphone schnell gemacht. Eine lokale KI verschlagwortet sie anhand des Inhalts – direkt auf dem NAS ohne leistungsfähige Grafikkarte.

Artikel verschenken

[In Pocket speichern](https://getpocket.com/save?url=https%3A%2F%2Fwww.heise.de%2Fratgeber%2FWie-eine-lokale-KI-die-Fotosammlung-auf-dem-NAS-verschlagworten-kann-9685509.html" \o "In Pocket speichern" \t "_blank) [vorlesen](https://app-eu.readspeaker.com/cgi-bin/rsent?customerid=4407&lang=de_de&readid=meldung&url=https%3A%2F%2Fwww.heise.de%2Fratgeber%2FWie-eine-lokale-KI-die-Fotosammlung-auf-dem-NAS-verschlagworten-kann-9685509.html%3Fseite%3Dall" \o "Beitrag vorlesen und MP3-Download" \t "_blank)

[Druckansicht](https://www.heise.de/ratgeber/Wie-eine-lokale-KI-die-Fotosammlung-auf-dem-NAS-verschlagworten-kann-9685509.html?view=print" \o "Druckansicht) [63 Kommentare lesen](https://www.heise.de/forum/heise-online/Kommentare/Wie-eine-lokale-KI-die-Fotosammlung-auf-dem-NAS-verschlagworten-kann/forum-539043/comment/" \o "Kommentar lesen)



(Bild: KI Midjourney | Collage c‘t)

22.04.2024, 10:30 Uhr

Lesezeit: 10 Min.

[c't Magazin](https://www.heise.de/ct/)

Von

* Ramon Wartala

Inhaltsverzeichnis

1. Wie eine lokale KI die Fotosammlung auf dem NAS verschlagworten kann
   * [Vielschichtig](https://www.heise.de/ratgeber/Wie-eine-lokale-KI-die-Fotosammlung-auf-dem-NAS-verschlagworten-kann-9685509.html?seite=all#nav_vielschichtig_0)
   * [KI als Container](https://www.heise.de/ratgeber/Wie-eine-lokale-KI-die-Fotosammlung-auf-dem-NAS-verschlagworten-kann-9685509.html?seite=all#nav_ki_als_container_1)
   * [Bilder verschlagworten](https://www.heise.de/ratgeber/Wie-eine-lokale-KI-die-Fotosammlung-auf-dem-NAS-verschlagworten-kann-9685509.html?seite=all#nav_bilder__2)
   * [Langsam aber stetig](https://www.heise.de/ratgeber/Wie-eine-lokale-KI-die-Fotosammlung-auf-dem-NAS-verschlagworten-kann-9685509.html?seite=all#nav_langsam_aber__3)
   * [Fazit](https://www.heise.de/ratgeber/Wie-eine-lokale-KI-die-Fotosammlung-auf-dem-NAS-verschlagworten-kann-9685509.html?seite=all#nav_fazit_4)

[Artikel in c't 10/2024 lesen](https://www.heise.de/select/ct/2024/10/2409912020727478813)

Seit jedes Smartphone eine Digitalkamera hat, wird so viel geknipst wie noch nie. Doch wer mistet noch regelmäßig die Bilderflut aus? Stattdessen landen die Fotos auf dem PC oder dem heimischen NAS, wenn der Speicher des Smartphones voll ist. Wenn man sie nicht eh gleich der Cloud überantwortet. Wer die Kontrolle über seine Daten behalten will, organisiert private Bilder deshalb lieber selbst. Doch in den Fotobergen das eine zu finden, das man gerade braucht, ist allein aufgrund der schieren Menge ein Problem. Und die Menge verhindert meist auch, die Fotos thematisch zu sortieren. Abhilfe verheißt eine KI, die Bilder analysiert und sie mit Schlagworten versieht.

Solche multimodale große Sprachmodelle (Large Language Models, LLM), die neben Text auch visuelle Inhalte verarbeiten können, bieten OpenAI, Google und anderen Firmen an, üblicherweise in einem Abo-Modell. Diese zu nutzen würde aber neben den Kosten wiederum bedeuten, die eigenen Bilder aus der Hand zu geben.

Mit LlaVA (Large Language Visual Assistant Modell) gibt es aber auch ein Open-Source-Modell, das aus einer Forschungskooperation von Microsoft mit den Universitäten Madison (Wisconsin) und Berkeley (Kalifornien) entstanden ist und das Sie vollständig lokal betreiben können.

c't kompakt

* Optimierte KI-Modelle liefern auch ohne Grafikkartenbeschleunigung Bildbeschreibungen in akzeptabler Zeit.
* Ollama als Docker-Container unterstützt viele Large Language Models, auch multimodale.
* Ein Python-Skript konvertiert die Bilder im Hintergrund selbständig.

**Vielschichtig**

Als multimodal bezeichnet man Modelle, die mehrere Arten digitaler Inhalte gemeinsam verarbeiten können. Damit sind sie in der Lage, auf inhaltsspezifische Prompts wie

Please scribe the following image in

any details. Make sure to be very

precise and take into account all the

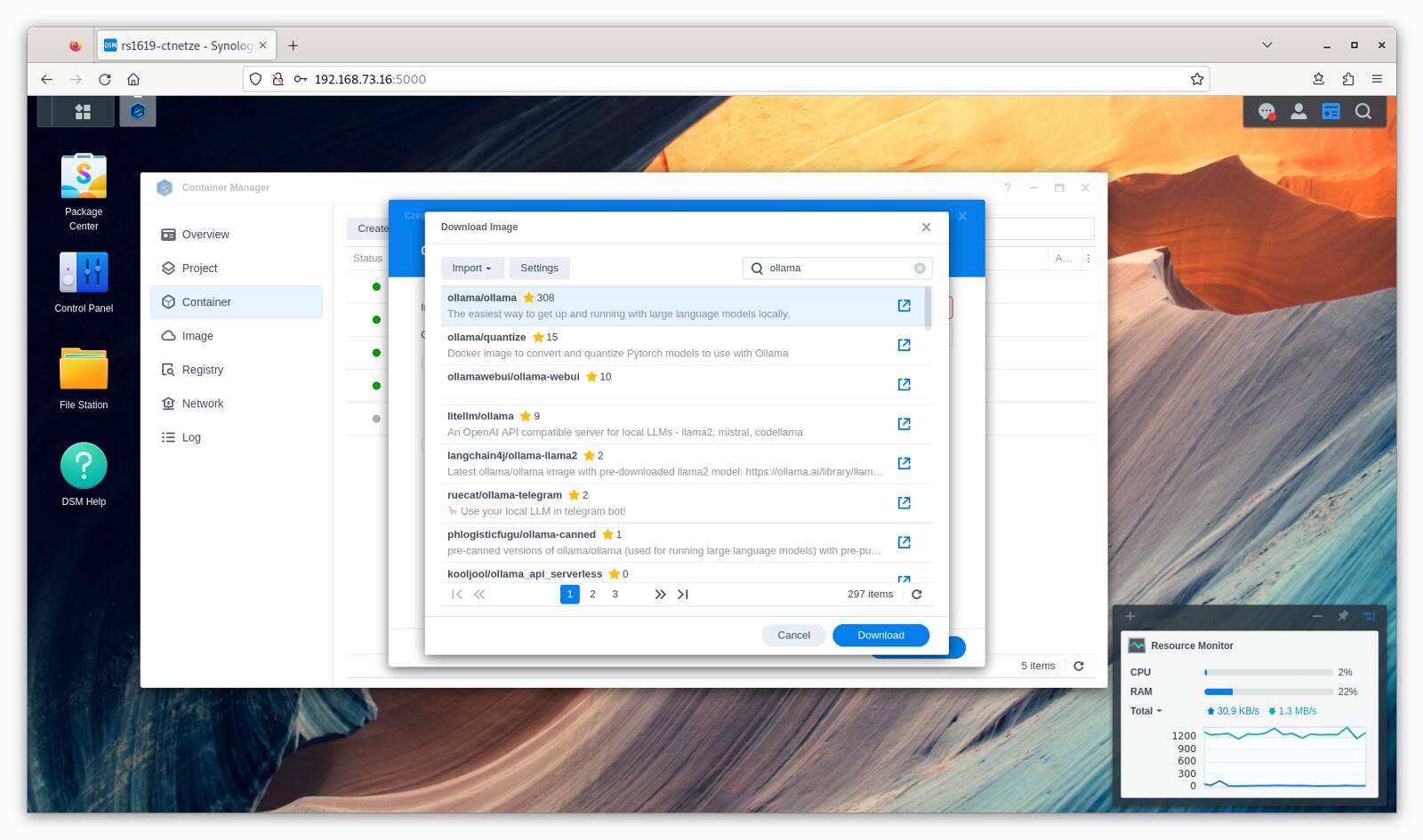
details such as colors and shapes

eine Antwort zu geben. Auch Texte in Bildern werden in Teilen erkannt, sodass sich auch einfache OCR-Aufgaben mit diesen Modellen erledigen lassen. Brauchte man dafür noch bis vor kurzem äußerst leistungsfähige Hardware wie zum Beispiel Grafikkarten von Nvidia, AMD oder bei Apple die M-Prozessoren, laufen diese Modelle dank der C/C++-Implementierung von Georgi Gerganov inzwischen auch auf leistungsschwächerer Hardware in erträglicher Geschwindigkeit.

Ein wesentlicher Punkt, die Hardwareanforderungen senken zu können, war die Quantisierung. Die Gewichte moderne LLM bestehen normalerweise aus 32-Bit- oder 16-Bit-Gleitkommazahlen, belegen also bis zu 4 Byte RAM. Das summiert sich selbst bei einem kleinen Modell wie LlaVA mit rund sieben Milliarden Gewichten auf 28 GByte RAM. Bei einer Quantisierung wird die Genauigkeit und damit der Speicherbedarf verringert, eine gängige Größe ist 4 Bit pro Gewicht. So schrumpft das Modell mit 7 Milliarden Gewichten auf nur noch 3,5 GByte zusammen und ist auch auf kleineren Systemen mit wenig Speicher nutzbar, beispielsweise einem Synology-NAS. Aufgrund der Größe der Modelle empfehlen wir 8 GByte RAM, der Autor hat eine entsprechend aufgerüstete DiskStation DS224+ verwendet – Modelle mit ARM-Prozessoren fallen aufgrund der Speicheranforderungen heraus.

**KI als Container**

Der einfachste Weg, das LlaVA-Modell auf einem Synology-NAS zu betreiben, ist der Container Manager für Docker im Management-GUI des NAS. Dazu empfehlen wir Ollama, das Sie in der Liste der aus dem Docker-Hub herunterladbaren Images unter "ollama/ollama" finden. Bei Ollama handelt es sich um ein generisches Image für verschiedene Large Language Models, darunter Llama2, CodeLlama, Mistral und Gemma. Bei den multimodalen Modellen unterstützt Ollama neben LlaVA auch BakLLava, das auf Mistral7B basiert. Außerdem können Sie eigene Modelle importieren.

[](https://www.heise.de/imgs/18/4/5/7/2/8/3/0/docker_create_container-e52474496c4481c9.png)

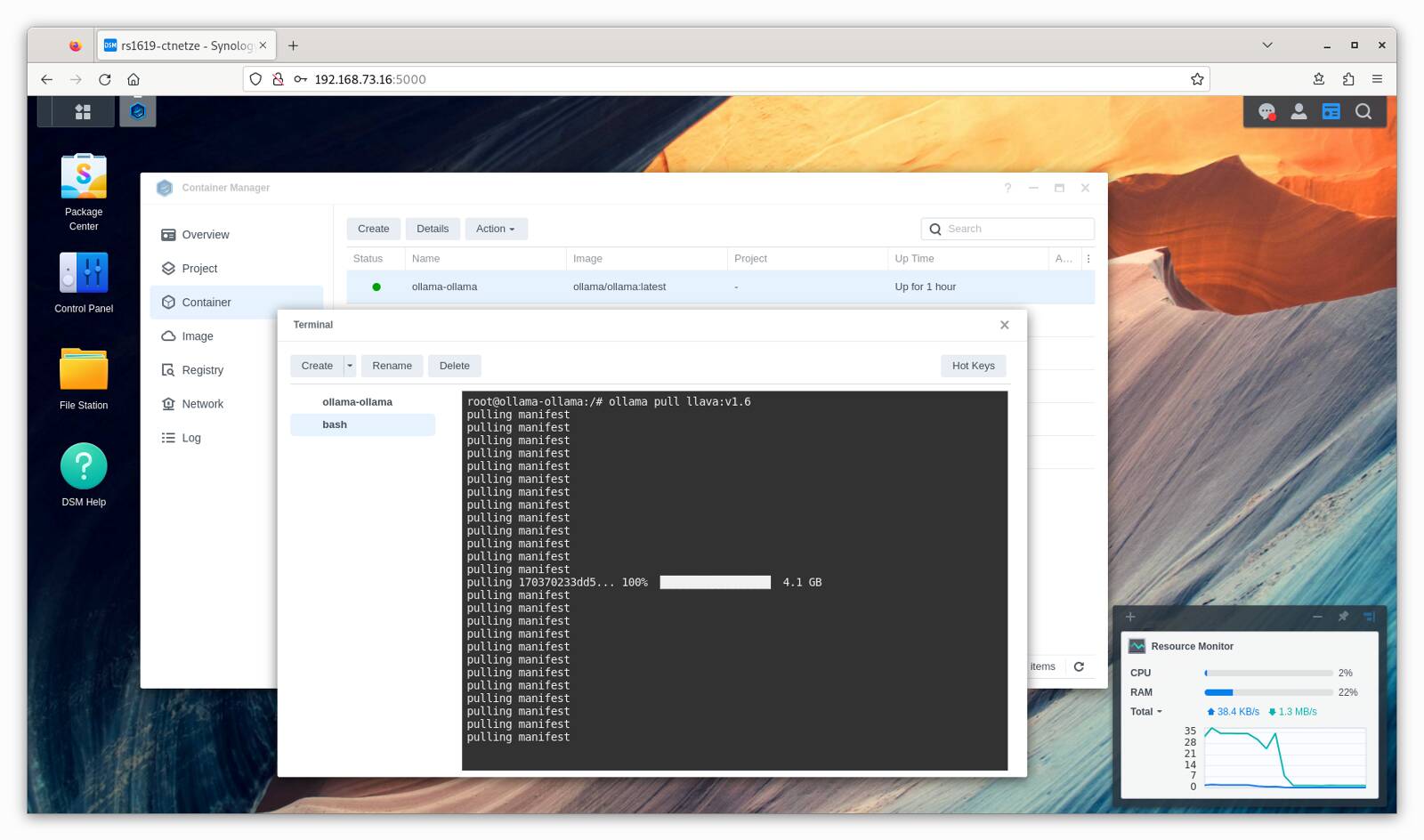
Das Docker-Image von Ollama kann viele verschiedene KI-Modelle beherbergen, auch das multimodale LlaVA, das Text und Bilder verarbeitet. Für die Kommunikation mit Anwendungen dient Port 11434.

Achten Sie bei der Einrichtung des Containers darauf, dass der Port 11434 erreichbar ist, denn darüber erfolgt anschließend die Kommunikation zwischen Client und Ollama. Idealerweise behalten Sie die Portnummer und auch den Containernamen "ollama" bei, davon gehen die nachfolgenden Beispiele jeweils aus. Frisch installiert hat Ollama keine Modelle an Bord, diese müssen Sie erst von Hand nachinstallieren. Dazu öffnen Sie im Container Manager aus dem Kontextmenü des Ollama-Containers das Terminal und legen mit einem Klick auf "Create" ein weiteres an, das den Namen "bash" erhält. Es handelt sich dabei um eine Root-Shell, in der Sie mit dem folgenden Befehl LlaVA Version 1.6 im Verzeichnis /root/.ollama installieren:

ollama pull llava:v1.6

Hat das geklappt, können Sie die Verbindung über das Client-API auf Port 11434 testen. Dazu loggen Sie sich per SSH auf Ihrem NAS ein und übermitteln mit curl eine Anfrage an das LlaVA-Modell:

curl http://127.0.0.1:11434/api/chat -d '{"model": "llava:v1.6", "messages": [ { "role": "user", "content": "Hi" } ] }'

[](https://www.heise.de/imgs/18/4/5/7/2/8/3/0/docker_terminal-0167bc2d269c6699.png)

Der Ollama-Container kommt ohne KI-Modell. Das müssen Sie über das Container-Terminal erst nachinstallieren.

Nach einigen Sekunden, in denen das Modell in den Speicher geladen wird, sollten Sie mehrere Zeilen als Antwort erhalten. Die letzte sollte folgendes Muster aufweisen:

{"model":"llava:v1.6",...,"message":{"role":"assistant","content":""},"done":true,...}

Damit wäre die KI einsatzbereit. Antwortet Ollama hingegen

{"error":"model 'llava:v1.6' not found, try pulling it first"}

so hat die Installation des Modells nicht geklappt und Sie müssen es noch einmal mittels ollama pull im Terminal des Containers versuchen. Bei weiteren Anfragen dauert es möglicherweise wieder einige Sekunden, bis das Modell reagiert ist, denn Ollama entlädt Modelle nach einigen Minuten wieder, wenn sie gerade nicht benutzt werden, um Speicher zu sparen.

**Bilder verschlagworten**

Um Ihre Bilder zu verschlagworten, haben wir Ihnen [das Python-Skript schlagwort.py zum Download bereitgestellt](https://ftp.heise.de/ct/listings/2024/10/schlagwort.py), das Sie direkt auf dem Synology-NAS einsetzen können. Es nutzt die Bibliothek LangChain, um das Client-API von Ollama anzusprechen, verkleinert die Originalbilder mittels Pillow, um sie an die KI zu übergeben, und fügt die Schlagworte schließlich mithilfe der Bibliothek IPTCinfo3 an das Originalbild an. Diese Bibliotheken sind standardmäßig jedoch nicht installiert, weshalb Sie zunächst die Python-Paketverwaltung pip und dann die Bibliotheken installieren müssen. Dies erledigen Sie mit den folgenden Befehlen via SSH-Verbindung zu Ihrem NAS:

python3 -m ensurepip --default-pip

python3 -m pip install --upgrade pip

python3 -m pip install langchain

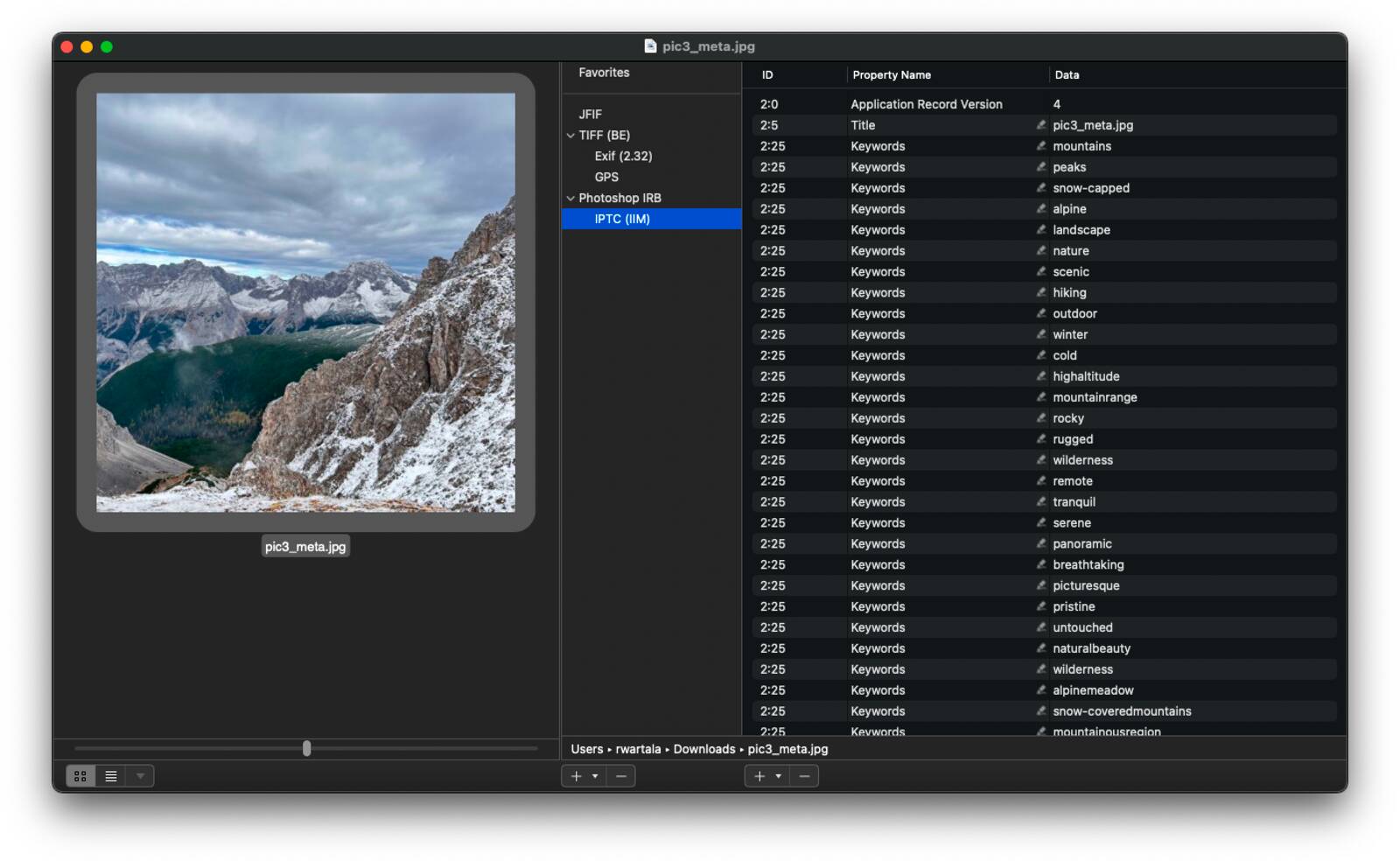
python3 -m pip install pillow

python3 -m pip install iptcinfo3

Anschließend können Sie die Verschlagwortung mit dem Befehl

python3 schlagwort.py

starten. Das Skript sucht standardmäßig im aktuellen Verzeichnis, in dem es aufgerufen wurde, nach Dateien mit der Endung ".jpg" und speichert deren verschlagwortete Versionen mit der Endung "\_meta.jpg". Wenn Sie stattdessen lieber ein freigegebenes Verzeichnis verwenden wollen, etwa /volume1/public/Fotos/input, so tragen Sie dieses in der Variablen SOURCE\_DIR am Anfang des Skripts ein. Über die Variable TARGET\_DIR gleich darunter bestimmen Sie, wo die verschlagworteten Bilder landen. Indem das Skript eine Kopie anlegt, verlieren Sie keine Daten, falls bei der Bildverarbeitung irgend etwas schiefgehen sollte, beispielsweise weil die Original-Bilddatei beschädigt war und Pillow sie deshalb nur teilweise einlesen konnte.

[](https://www.heise.de/imgs/18/4/5/7/2/8/3/0/iptc_keywords_photometaedit-b789c0ea7c9f7209.png)

Die Schlagworte speichert das Python-Skript unter keywords in den Metadaten des Bilds, hier die Übersicht aus Photo Meta Edit für macOS.

Um LlaVA Schlagworte für die Bilder zu entlocken, verwendet das Skript das Prompt "Please find very precise keywords and separate them with commas." Das Ergebnis speichert es im Feld keywords der IPTC-Metadaten, außerdem den Namen des Skripts im Feld writer/editor, sodass sich später nachvollziehen lässt, woher die Schlagworte stammen. Eine solche Verschlagwortung hilft nicht nur, einzelne Bilder in riesigen Fotoalben wieder aufzuspüren, sondern auch, um das SEO-Ranking einer Website zu verbessern. Denn Google und andere Suchmaschinen bevorzugen Websites mit Bildern, die mit Zusatzinformationen bestückt sind.

**Langsam aber stetig**

Ohne Unterstützung einer GPU vergeht allerdings einige Zeit, bis Ihre Bilder verschlagwortet sind. Auf der DiskStation DS224+ mit dem mobilen Vierkern-Prozessor Intel Celeron J4125 dauerte es rund 20 Minuten pro Bild, bis LlaVA die Ergebnisse lieferte. Die leistungsfähigere RackStation RS1619xs+ mit Intels älterem Quad-Core Xeon D-1527 aus dem Serversegment war schon nach rund 5 Minuten fertig. Um ein ganzes Verzeichnis voller Bilder zu verarbeiten, sollte das Python-Skript deshalb im Hintergrund weiter laufen, ohne dass Sie die SSH-Verbindung offen halten müssen.

Dazu eignet sich das Kommando screen: Es öffnet eine Shell, die von der gerade laufenden Terminal-Session abgekoppelt ist. Der Aufruf von python3 schlagwort.py & hingegen würde nur einen Hintergrundprozess im aktuellen Terminal starten, der beendet wird, wenn Sie die SSH-Sitzung schließen. Eine Screen-Session hingegen können Sie jederzeit mit der Tastenkombination Strg+A und dann der Taste D von der Terminal-Session abkoppeln und im Hintergrund weiter ausführen lassen. Das Kommando, das in der Screen-Session ausgeführt werden soll, hängen Sie einfach an:

screen python3 schlagwort.py

Wollen Sie später nach dem aktuellen Stand der Verarbeitung schauen, können Sie die abgekoppelte Screen-Session mit dem Befehl screen -r wieder hervorholen und bei Bedarf erneut per Tastatur mit Strg+A, D in den Hintergrund schicken. Gibt es mehrere Screen-Sessions im Hintergrund, so zeigt sie Ihnen screen -ls als Liste; hinter dem Parameter -r können Sie angeben, an welche der Sessions Sie anknüpfen wollen.

Ist die Bildverarbeitung abgeschlossen, so endet auch die Screen-Session automatisch. Finden Sie also eine zuvor in den Hintergrund verlegte Screen-Session nicht mehr, so bedeutet das nur, dass die Arbeit bereits erledigt wurde.

Ganz ausgereift ist die Lösung noch nicht. Zunächst einmal liefert LlaVA lediglich englischsprachige Schlagworte, eine Übersetzungsautomatik gibt es nicht. Sie müssen künftig also auch englische Suchbegriffe verwenden. Außerdem prüft schlagwort.py noch nicht, ob ein Bild bereits verarbeitet wurde, sodass Sie das Skript derzeit nicht automatisch etwa per Cron-Job auf dem NAS ausführen lassen können.

**Fazit**

Indem Sie schlagwort.py Verzeichnis für Verzeichnis verarbeiten lassen, können Sie über die Zeit auch größere Fotosammlungen direkt auf dem NAS verschlagworten. Bei neuen Bildern sollten Sie sich merken, sie zunächst in das in schlagwort.py unter input definierte Verzeichnis zu verschieben und das Skript dann zu starten.

Tags darauf können Sie die Bilder dann in Ihre Fotosammlung verschieben und künftig leichter wiederfinden. Und das alles, ohne Ihre privaten Daten in eine Cloud oder an eine externe Stelle hochladen zu müssen. Die Kontrolle über die Verbreitung liegt weiterhin ganz bei Ihnen. ([mid](mailto:mid@ct.de" \o "Mirko Dölle))