

BitePulse AI - Dataset acquisition and audit

Introduction

Here in this notebook, we'll download, verify, and organize the EatSense dataset for BitePulse AI:

- Mount Google Drive, confirm free space, and set a single data root.
- Download EatSense data using a fast, resumable method.
- Verify archives for integrity before extraction.
- Extract archives into organized folders, including any zip files packed inside other zips.
- Generate a quick inventory of folders, file counts, and sizes.
- Sanity check the setup.

Folders Plan:

- `.../eatsense/zips` holds the original downloads.
- `.../eatsense/rgb` holds RGB videos.
- `.../eatsense/depth` holds depth maps.
- `.../eatsense/poses_true` holds true face pose csv files.
- `.../eatsense/poses_fake` holds fake face pose csv files.
- `.../eatsense/test` holds the official test set.
- `.../eatsense/all_misc` holds documents and readme files

Dataset

BitePulse AI is trained on [EatSense](#), a research dataset of real people eating real meals, with every frame labeled for actions like eating, chewing, drinking, or resting. Faces are anonymized. This lets us train deep learning models to detect bites and pacing patterns without recording new users.

In the real product, BitePulse AI can run privately on device for consumers, or through a secure backend for wellness partners. For this capstone, we only use the public [EatSense](#) dataset, and we do not collect or store new personal video.

Imports and basic setup

```
# Due to EatSense Large files let's download Faster downloader aria2:  
!apt-get -y install -qq aria2
```

```
Selecting previously unselected package libc-ares2:amd64.  
(Reading database ... 125079 files and directories currently  
installed.)
```

```
Preparing to unpack .../libc-ares2_1.18.1-  
1ubuntu0.22.04.3_amd64.deb ...
```

```
Unpacking libc-ares2:amd64 (1.18.1-1ubuntu0.22.04.3) ...
```

```
Selecting previously unselected package libaria2-0:amd64.  
Preparing to unpack .../libaria2-0_1.36.0-1_amd64.deb ...  
Unpacking libaria2-0:amd64 (1.36.0-1) ...  
Selecting previously unselected package aria2.  
Preparing to unpack .../aria2_1.36.0-1_amd64.deb ...  
Unpacking aria2 (1.36.0-1) ...  
Setting up libc-ares2:amd64 (1.18.1-1ubuntu0.22.04.3) ...  
Setting up libaria2-0:amd64 (1.36.0-1) ...  
Setting up aria2 (1.36.0-1) ...  
Processing triggers for man-db (2.10.2-1) ...  
Processing triggers for libc-bin (2.35-0ubuntu3.8) ...  
/sbin/ldconfig.real: /usr/local/lib/libtbbbind_2_5.so.3 is not a  
symbolic link  
  
/sbin/ldconfig.real: /usr/local/lib/libur_loader.so.0 is not a  
symbolic link  
  
/sbin/ldconfig.real: /usr/local/lib/libtbbbind_2_0.so.3 is not a  
symbolic link  
  
/sbin/ldconfig.real: /usr/local/lib/libur_adapter_opencl.so.0 is not a  
symbolic link  
  
/sbin/ldconfig.real: /usr/local/lib/libtbb.so.12 is not a symbolic  
link  
  
/sbin/ldconfig.real: /usr/local/lib/libtbbmalloc_proxy.so.2 is not a  
symbolic link  
  
/sbin/ldconfig.real: /usr/local/lib/libur_adapter_level_zero.so.0 is  
not a symbolic link  
  
/sbin/ldconfig.real: /usr/local/lib/libur_adapter_level_zero_v2.so.0  
is not a symbolic link  
  
/sbin/ldconfig.real: /usr/local/lib/libumf.so.1 is not a symbolic link  
  
/sbin/ldconfig.real: /usr/local/lib/libtbbmalloc.so.2 is not a  
symbolic link  
  
/sbin/ldconfig.real: /usr/local/lib/libhwloc.so.15 is not a symbolic  
link  
  
/sbin/ldconfig.real: /usr/local/lib/libtcm_debug.so.1 is not a  
symbolic link  
  
/sbin/ldconfig.real: /usr/local/lib/libtcm.so.1 is not a symbolic link
```

```
# Mount Google Drive:
from google.colab import drive
drive.mount('/content/drive')

Mounted at /content/drive

import glob, os, shlex, subprocess, collections, re
from pathlib import Path
import pandas as pd
import numpy as np
import cv2
from google.colab.patches import cv2_imshow
```

Setups the Google drives paths

```
# Paths:
ROOT = Path("/content/drive/MyDrive/eatsense")
ZIPS = ROOT / "zips"
OUT = ROOT

RGB_DIR = OUT / "rgb"
DEPTH_DIR = OUT / "depth"
POSES_TRUE_DIR = OUT / "poses_true"
POSES_FAKE_DIR = OUT / "poses_fake"
TEST_DIR = OUT / "test"
MISC_DIR = OUT / "all_misc"

# Create folders if missing:
for p in [ZIPS, RGB_DIR, DEPTH_DIR, POSES_TRUE_DIR, POSES_FAKE_DIR,
TEST_DIR, MISC_DIR]:
    p.mkdir(parents=True, exist_ok=True)

# Make paths available to bash cells:
os.environ["ROOT"] = str(ROOT)
os.environ["ZIPS"] = str(ZIPS)
os.environ["OUT"] = str(OUT)

# Quick environment summary:
print("Root:", ROOT)
print("Zips:", ZIPS)
print("Output:", OUT)

Root: /content/drive/MyDrive/eatsense
Zips: /content/drive/MyDrive/eatsense/zips
Output: /content/drive/MyDrive/eatsense

# Free space check:
!df -h /content/drive | tail -n 1

drive          100G   3.9G   97G    4% /content/drive
```

Download the dataset- EatSense

Here let's download the EatSense archives into the zips folder on Drive. I will use aria2 for resumable downloads. I will start with the All in one zip. Later will be adding the individual archives.

```
# Alright now let's use All in one zip URL from the EatSense page:
ALL_URL = "https://datashare.ed.ac.uk/download/DS_10283_8759.zip"
```

```
os.environ["ALL_URL"] = ALL_URL
```

```
# Confirm:
```

```
print("ALL_URL:", os.environ["ALL_URL"])
print("ZIPS:", os.environ["ZIPS"])
```

```
ALL_URL: https://datashare.ed.ac.uk/download/DS_10283_8759.zip
ZIPS: /content/drive/MyDrive/eatsense/zips
```

```
%%bash
```

```
# Download to the zips folder:
```

```
aria2c -c -x 16 -s 16 -d "$ZIPS" -o "eatsense_all_in_one.zip"
"$ALL_URL"
```

```
10/31 02:20:26 [NOTICE] Downloading 1 item(s)
```

```
10/31 02:20:26 [NOTICE] Allocating disk space. Use --file-
allocation=none to disable it. See --file-allocation option in man
page for more details.
```

```
[#c4955a 0B/27GiB(0%) CN:1 DL:0B] [FileAlloc:#c4955a 307MiB/27GiB(1%)]
[#c4955a 0B/27GiB(0%) CN:1 DL:0B] [FileAlloc:#c4955a 0.9GiB/27GiB(3%)]
[#c4955a 0B/27GiB(0%) CN:1 DL:0B] [FileAlloc:#c4955a 1.4GiB/27GiB(5%)]
[#c4955a 0B/27GiB(0%) CN:1 DL:0B] [FileAlloc:#c4955a 2.1GiB/27GiB(7%)]
[#c4955a 0B/27GiB(0%) CN:1 DL:0B] [FileAlloc:#c4955a 2.3GiB/27GiB(8%)]
[#c4955a 0B/27GiB(0%) CN:1 DL:0B] [FileAlloc:#c4955a 2.5GiB/27GiB(9%)]
[#c4955a 0B/27GiB(0%) CN:1 DL:0B] [FileAlloc:#c4955a 2.7GiB/27GiB(9%)]
[#c4955a 0B/27GiB(0%) CN:1 DL:0B] [FileAlloc:#c4955a
3.1GiB/27GiB(11%)]
[#c4955a 0B/27GiB(0%) CN:1 DL:0B] [FileAlloc:#c4955a
3.4GiB/27GiB(12%)]
[#c4955a 0B/27GiB(0%) CN:1 DL:0B] [FileAlloc:#c4955a
4.0GiB/27GiB(14%)]
[#c4955a 0B/27GiB(0%) CN:1 DL:0B] [FileAlloc:#c4955a
4.6GiB/27GiB(17%)]
[#c4955a 0B/27GiB(0%) CN:1 DL:0B] [FileAlloc:#c4955a
5.2GiB/27GiB(19%)]
[#c4955a 0B/27GiB(0%) CN:1 DL:0B] [FileAlloc:#c4955a
5.6GiB/27GiB(20%)]
[#c4955a 0B/27GiB(0%) CN:1 DL:0B] [FileAlloc:#c4955a
```

```
5.7GiB/27GiB(20%)]
[#c4955a 0B/27GiB(0%) CN:1 DL:0B] [FileAlloc:#c4955a
6.3GiB/27GiB(23%)]
[#c4955a 0B/27GiB(0%) CN:1 DL:0B] [FileAlloc:#c4955a
6.9GiB/27GiB(25%)]
[#c4955a 0B/27GiB(0%) CN:1 DL:0B] [FileAlloc:#c4955a
7.4GiB/27GiB(27%)]
[#c4955a 0B/27GiB(0%) CN:1 DL:0B] [FileAlloc:#c4955a
7.8GiB/27GiB(28%)]
[#c4955a 0B/27GiB(0%) CN:1 DL:0B] [FileAlloc:#c4955a
7.8GiB/27GiB(28%)]
[#c4955a 0B/27GiB(0%) CN:1 DL:0B] [FileAlloc:#c4955a
7.8GiB/27GiB(28%)]
[#c4955a 0B/27GiB(0%) CN:1 DL:0B] [FileAlloc:#c4955a
8.5GiB/27GiB(30%)]
[#c4955a 0B/27GiB(0%) CN:1 DL:0B] [FileAlloc:#c4955a
9.0GiB/27GiB(32%)]
[#c4955a 0B/27GiB(0%) CN:1 DL:0B] [FileAlloc:#c4955a
9.3GiB/27GiB(33%)]
[#c4955a 0B/27GiB(0%) CN:1 DL:0B] [FileAlloc:#c4955a
9.4GiB/27GiB(34%)]
[#c4955a 0B/27GiB(0%) CN:1 DL:0B] [FileAlloc:#c4955a
9.9GiB/27GiB(36%)]
[#c4955a 0B/27GiB(0%) CN:1 DL:0B] [FileAlloc:#c4955a 10GiB/27GiB(38%)]
[#c4955a 0B/27GiB(0%) CN:1 DL:0B] [FileAlloc:#c4955a 10GiB/27GiB(39%)]
[#c4955a 0B/27GiB(0%) CN:1 DL:0B] [FileAlloc:#c4955a 11GiB/27GiB(41%)]
[#c4955a 0B/27GiB(0%) CN:1 DL:0B] [FileAlloc:#c4955a 11GiB/27GiB(43%)]
[#c4955a 0B/27GiB(0%) CN:1 DL:0B] [FileAlloc:#c4955a 12GiB/27GiB(45%)]
[#c4955a 0B/27GiB(0%) CN:1 DL:0B] [FileAlloc:#c4955a 12GiB/27GiB(46%)]
[#c4955a 0B/27GiB(0%) CN:1 DL:0B] [FileAlloc:#c4955a 12GiB/27GiB(47%)]
[#c4955a 0B/27GiB(0%) CN:1 DL:0B] [FileAlloc:#c4955a 13GiB/27GiB(47%)]
[#c4955a 0B/27GiB(0%) CN:1 DL:0B] [FileAlloc:#c4955a 13GiB/27GiB(49%)]
[#c4955a 0B/27GiB(0%) CN:1 DL:0B] [FileAlloc:#c4955a 14GiB/27GiB(51%)]
[#c4955a 0B/27GiB(0%) CN:1 DL:0B] [FileAlloc:#c4955a 14GiB/27GiB(53%)]
[#c4955a 0B/27GiB(0%) CN:1 DL:0B] [FileAlloc:#c4955a 15GiB/27GiB(55%)]
[#c4955a 0B/27GiB(0%) CN:1 DL:0B] [FileAlloc:#c4955a 15GiB/27GiB(57%)]
[#c4955a 0B/27GiB(0%) CN:1 DL:0B] [FileAlloc:#c4955a 16GiB/27GiB(59%)]
[#c4955a 0B/27GiB(0%) CN:1 DL:0B] [FileAlloc:#c4955a 16GiB/27GiB(61%)]
[#c4955a 0B/27GiB(0%) CN:1 DL:0B] [FileAlloc:#c4955a 17GiB/27GiB(62%)]
[#c4955a 0B/27GiB(0%) CN:1 DL:0B] [FileAlloc:#c4955a 17GiB/27GiB(62%)]
[#c4955a 0B/27GiB(0%) CN:1 DL:0B] [FileAlloc:#c4955a 17GiB/27GiB(64%)]
[#c4955a 0B/27GiB(0%) CN:1 DL:0B] [FileAlloc:#c4955a 18GiB/27GiB(67%)]
[#c4955a 0B/27GiB(0%) CN:1 DL:0B] [FileAlloc:#c4955a 18GiB/27GiB(68%)]
[#c4955a 0B/27GiB(0%) CN:1 DL:0B] [FileAlloc:#c4955a 19GiB/27GiB(69%)]
[#c4955a 0B/27GiB(0%) CN:1 DL:0B] [FileAlloc:#c4955a 19GiB/27GiB(70%)]
[#c4955a 0B/27GiB(0%) CN:1 DL:0B] [FileAlloc:#c4955a 19GiB/27GiB(71%)]
[#c4955a 0B/27GiB(0%) CN:1 DL:0B] [FileAlloc:#c4955a 20GiB/27GiB(73%)]
[#c4955a 0B/27GiB(0%) CN:1 DL:0B] [FileAlloc:#c4955a 20GiB/27GiB(75%)]
```

*** Download Progress Summary as of Fri Oct 31 02:21:26 2025 ***

```
=====
=====
[#c4955a 0B/27GiB(0%) CN:1 DL:0B]
FILE: /content/drive/MyDrive/eatsense/zips/eatsense_all_in_one.zip
-----
-----

[#c4955a 0B/27GiB(0%) CN:1 DL:0B] [FileAlloc:#c4955a 21GiB/27GiB(76%)]
[#c4955a 0B/27GiB(0%) CN:1 DL:0B] [FileAlloc:#c4955a 21GiB/27GiB(76%)]
[#c4955a 0B/27GiB(0%) CN:1 DL:0B] [FileAlloc:#c4955a 21GiB/27GiB(78%)]
[#c4955a 0B/27GiB(0%) CN:1 DL:0B] [FileAlloc:#c4955a 22GiB/27GiB(80%)]
[#c4955a 0B/27GiB(0%) CN:1 DL:0B] [FileAlloc:#c4955a 22GiB/27GiB(82%)]
[#c4955a 0B/27GiB(0%) CN:1 DL:0B] [FileAlloc:#c4955a 23GiB/27GiB(83%)]
[#c4955a 0B/27GiB(0%) CN:1 DL:0B] [FileAlloc:#c4955a 23GiB/27GiB(84%)]
[#c4955a 0B/27GiB(0%) CN:1 DL:0B] [FileAlloc:#c4955a 23GiB/27GiB(85%)]
[#c4955a 0B/27GiB(0%) CN:1 DL:0B] [FileAlloc:#c4955a 23GiB/27GiB(86%)]
[#c4955a 0B/27GiB(0%) CN:1 DL:0B] [FileAlloc:#c4955a 24GiB/27GiB(88%)]
[#c4955a 0B/27GiB(0%) CN:1 DL:0B] [FileAlloc:#c4955a 24GiB/27GiB(90%)]
[#c4955a 0B/27GiB(0%) CN:1 DL:0B] [FileAlloc:#c4955a 25GiB/27GiB(91%)]
[#c4955a 0B/27GiB(0%) CN:1 DL:0B] [FileAlloc:#c4955a 25GiB/27GiB(92%)]
[#c4955a 0B/27GiB(0%) CN:1 DL:0B] [FileAlloc:#c4955a 25GiB/27GiB(93%)]
[#c4955a 0B/27GiB(0%) CN:1 DL:0B] [FileAlloc:#c4955a 26GiB/27GiB(95%)]
[#c4955a 0B/27GiB(0%) CN:1 DL:0B] [FileAlloc:#c4955a 26GiB/27GiB(97%)]
[#c4955a 0B/27GiB(0%) CN:1 DL:0B] [FileAlloc:#c4955a 27GiB/27GiB(98%)]
[#c4955a 0B/27GiB(0%) CN:1 DL:0B]
[#c4955a 3.1MiB/27GiB(0%) CN:16 DL:3.3MiB ETA:2h21m13s]
[#c4955a 23MiB/27GiB(0%) CN:16 DL:11MiB ETA:40m9s]
[#c4955a 72MiB/27GiB(0%) CN:16 DL:24MiB ETA:19m16s]
[#c4955a 145MiB/27GiB(0%) CN:16 DL:36MiB ETA:12m44s]
[#c4955a 213MiB/27GiB(0%) CN:16 DL:42MiB ETA:10m51s]
[#c4955a 281MiB/27GiB(1%) CN:16 DL:47MiB ETA:9m49s]
[#c4955a 355MiB/27GiB(1%) CN:16 DL:50MiB ETA:9m4s]
[#c4955a 428MiB/27GiB(1%) CN:16 DL:53MiB ETA:8m34s]
[#c4955a 483MiB/27GiB(1%) CN:16 DL:53MiB ETA:8m32s]
[#c4955a 552MiB/27GiB(1%) CN:16 DL:55MiB ETA:8m16s]
[#c4955a 609MiB/27GiB(2%) CN:16 DL:60MiB ETA:7m31s]
[#c4955a 673MiB/27GiB(2%) CN:16 DL:65MiB ETA:6m59s]
[#c4955a 730MiB/27GiB(2%) CN:16 DL:66MiB ETA:6m54s]
[#c4955a 795MiB/27GiB(2%) CN:16 DL:65MiB ETA:6m58s]
[#c4955a 862MiB/27GiB(3%) CN:16 DL:65MiB ETA:6m58s]
[#c4955a 0.9GiB/27GiB(3%) CN:16 DL:65MiB ETA:6m55s]
[#c4955a 0.9GiB/27GiB(3%) CN:16 DL:59MiB ETA:7m37s]
[#c4955a 1.0GiB/27GiB(3%) CN:16 DL:60MiB ETA:7m26s]
[#c4955a 1.0GiB/27GiB(3%) CN:16 DL:61MiB ETA:7m19s]
[#c4955a 1.1GiB/27GiB(4%) CN:16 DL:63MiB ETA:7m6s]
[#c4955a 1.2GiB/27GiB(4%) CN:16 DL:64MiB ETA:6m54s]
[#c4955a 1.3GiB/27GiB(4%) CN:16 DL:67MiB ETA:6m38s]
[#c4955a 1.3GiB/27GiB(5%) CN:16 DL:68MiB ETA:6m31s]
[#c4955a 1.4GiB/27GiB(5%) CN:16 DL:68MiB ETA:6m29s]
```

```
[#c4955a 1.5GiB/27GiB(5%) CN:16 DL:69MiB ETA:6m23s]
[#c4955a 1.6GiB/27GiB(5%) CN:16 DL:74MiB ETA:5m53s]
[#c4955a 1.6GiB/27GiB(6%) CN:16 DL:74MiB ETA:5m53s]
[#c4955a 1.7GiB/27GiB(6%) CN:16 DL:74MiB ETA:5m51s]
[#c4955a 1.8GiB/27GiB(6%) CN:16 DL:74MiB ETA:5m51s]
[#c4955a 1.8GiB/27GiB(6%) CN:16 DL:74MiB ETA:5m52s]
[#c4955a 1.9GiB/27GiB(7%) CN:16 DL:73MiB ETA:5m54s]
[#c4955a 2.0GiB/27GiB(7%) CN:16 DL:73MiB ETA:5m53s]
[#c4955a 2.1GiB/27GiB(7%) CN:16 DL:73MiB ETA:5m52s]
[#c4955a 2.1GiB/27GiB(7%) CN:16 DL:73MiB ETA:5m52s]
[#c4955a 2.2GiB/27GiB(8%) CN:16 DL:73MiB ETA:5m51s]
[#c4955a 2.3GiB/27GiB(8%) CN:16 DL:73MiB ETA:5m50s]
[#c4955a 2.3GiB/27GiB(8%) CN:16 DL:73MiB ETA:5m47s]
```

*** Download Progress Summary as of Fri Oct 31 02:22:27 2025 ***

```
=====
[ #c4955a 2.4GiB/27GiB(8%) CN:16 DL:72MiB ETA:5m54s]
FILE: /content/drive/MyDrive/eatsense/zips/eatsense_all_in_one.zip
-----
-----
```

```
[#c4955a 2.4GiB/27GiB(8%) CN:16 DL:72MiB ETA:5m54s]
[#c4955a 2.5GiB/27GiB(9%) CN:16 DL:71MiB ETA:5m57s]
[#c4955a 2.5GiB/27GiB(9%) CN:16 DL:71MiB ETA:5m56s]
[#c4955a 2.6GiB/27GiB(9%) CN:16 DL:71MiB ETA:5m57s]
[#c4955a 2.7GiB/27GiB(9%) CN:16 DL:69MiB ETA:6m6s]
[#c4955a 2.7GiB/27GiB(10%) CN:16 DL:67MiB ETA:6m14s]
[#c4955a 2.8GiB/27GiB(10%) CN:16 DL:67MiB ETA:6m15s]
[#c4955a 2.9GiB/27GiB(10%) CN:16 DL:67MiB ETA:6m14s]
[#c4955a 2.9GiB/27GiB(10%) CN:16 DL:67MiB ETA:6m14s]
[#c4955a 3.0GiB/27GiB(11%) CN:16 DL:67MiB ETA:6m11s]
[#c4955a 3.1GiB/27GiB(11%) CN:16 DL:68MiB ETA:6m2s]
[#c4955a 3.1GiB/27GiB(11%) CN:16 DL:70MiB ETA:5m54s]
[#c4955a 3.2GiB/27GiB(11%) CN:16 DL:70MiB ETA:5m49s]
[#c4955a 3.3GiB/27GiB(12%) CN:16 DL:72MiB ETA:5m42s]
[#c4955a 3.4GiB/27GiB(12%) CN:16 DL:74MiB ETA:5m30s]
[#c4955a 3.4GiB/27GiB(12%) CN:16 DL:73MiB ETA:5m32s]
[#c4955a 3.5GiB/27GiB(12%) CN:16 DL:74MiB ETA:5m27s]
[#c4955a 3.6GiB/27GiB(13%) CN:16 DL:74MiB ETA:5m28s]
[#c4955a 3.7GiB/27GiB(13%) CN:16 DL:74MiB ETA:5m26s]
[#c4955a 3.7GiB/27GiB(13%) CN:16 DL:74MiB ETA:5m25s]
[#c4955a 3.8GiB/27GiB(14%) CN:16 DL:74MiB ETA:5m25s]
[#c4955a 3.9GiB/27GiB(14%) CN:16 DL:72MiB ETA:5m30s]
[#c4955a 3.9GiB/27GiB(14%) CN:16 DL:72MiB ETA:5m29s]
[#c4955a 4.0GiB/27GiB(14%) CN:16 DL:72MiB ETA:5m28s]
[#c4955a 4.1GiB/27GiB(15%) CN:16 DL:73MiB ETA:5m26s]
[#c4955a 4.2GiB/27GiB(15%) CN:16 DL:74MiB ETA:5m21s]
[#c4955a 4.2GiB/27GiB(15%) CN:16 DL:74MiB ETA:5m20s]
[#c4955a 4.3GiB/27GiB(15%) CN:16 DL:74MiB ETA:5m19s]
```

```
[#c4955a 4.4GiB/27GiB(16%) CN:16 DL:72MiB ETA:5m26s]
[#c4955a 4.4GiB/27GiB(16%) CN:16 DL:71MiB ETA:5m28s]
[#c4955a 4.5GiB/27GiB(16%) CN:16 DL:72MiB ETA:5m24s]
[#c4955a 4.6GiB/27GiB(16%) CN:16 DL:72MiB ETA:5m21s]
[#c4955a 4.6GiB/27GiB(17%) CN:16 DL:72MiB ETA:5m19s]
[#c4955a 4.7GiB/27GiB(17%) CN:16 DL:73MiB ETA:5m18s]
[#c4955a 4.8GiB/27GiB(17%) CN:16 DL:72MiB ETA:5m18s]
[#c4955a 4.9GiB/27GiB(17%) CN:16 DL:73MiB ETA:5m15s]
[#c4955a 4.9GiB/27GiB(18%) CN:16 DL:72MiB ETA:5m16s]
[#c4955a 5.0GiB/27GiB(18%) CN:16 DL:74MiB ETA:5m8s]
[#c4955a 5.1GiB/27GiB(18%) CN:16 DL:75MiB ETA:5m3s]
[#c4955a 5.2GiB/27GiB(18%) CN:16 DL:75MiB ETA:5m1s]
[#c4955a 5.2GiB/27GiB(19%) CN:16 DL:75MiB ETA:5m]
[#c4955a 5.3GiB/27GiB(19%) CN:16 DL:74MiB ETA:5m2s]
[#c4955a 5.4GiB/27GiB(19%) CN:16 DL:73MiB ETA:5m6s]
[#c4955a 5.4GiB/27GiB(19%) CN:16 DL:72MiB ETA:5m9s]
[#c4955a 5.5GiB/27GiB(20%) CN:16 DL:73MiB ETA:5m7s]
[#c4955a 5.6GiB/27GiB(20%) CN:16 DL:72MiB ETA:5m7s]
[#c4955a 5.7GiB/27GiB(20%) CN:16 DL:73MiB ETA:5m4s]
[#c4955a 5.7GiB/27GiB(21%) CN:16 DL:73MiB ETA:5m3s]
[#c4955a 5.8GiB/27GiB(21%) CN:16 DL:73MiB ETA:5m1s]
[#c4955a 5.9GiB/27GiB(21%) CN:16 DL:73MiB ETA:5m]
[#c4955a 5.9GiB/27GiB(21%) CN:16 DL:73MiB ETA:4m58s]
[#c4955a 6.0GiB/27GiB(22%) CN:16 DL:74MiB ETA:4m53s]
[#c4955a 6.1GiB/27GiB(22%) CN:16 DL:75MiB ETA:4m48s]
[#c4955a 6.2GiB/27GiB(22%) CN:16 DL:75MiB ETA:4m47s]
[#c4955a 6.2GiB/27GiB(22%) CN:16 DL:75MiB ETA:4m48s]
[#c4955a 6.3GiB/27GiB(23%) CN:16 DL:74MiB ETA:4m49s]
[#c4955a 6.4GiB/27GiB(23%) CN:16 DL:74MiB ETA:4m47s]
[#c4955a 6.5GiB/27GiB(23%) CN:16 DL:74MiB ETA:4m46s]
[#c4955a 6.5GiB/27GiB(23%) CN:16 DL:74MiB ETA:4m45s]
[#c4955a 6.6GiB/27GiB(24%) CN:16 DL:74MiB ETA:4m45s]
```

*** Download Progress Summary as of Fri Oct 31 02:23:27 2025 ***

```
=====
=====
```

```
[#c4955a 6.6GiB/27GiB(24%) CN:16 DL:68MiB ETA:5m9s]
FILE: /content/drive/MyDrive/eatsense/zips/eatsense_all_in_one.zip
-----
-----
```

```
[#c4955a 6.6GiB/27GiB(24%) CN:16 DL:68MiB ETA:5m9s]
[#c4955a 6.7GiB/27GiB(24%) CN:16 DL:69MiB ETA:5m5s]
[#c4955a 6.8GiB/27GiB(24%) CN:16 DL:69MiB ETA:5m4s]
[#c4955a 6.9GiB/27GiB(25%) CN:16 DL:69MiB ETA:5m3s]
[#c4955a 6.9GiB/27GiB(25%) CN:16 DL:68MiB ETA:5m5s]
[#c4955a 7.0GiB/27GiB(25%) CN:16 DL:67MiB ETA:5m10s]
[#c4955a 7.1GiB/27GiB(25%) CN:16 DL:66MiB ETA:5m11s]
[#c4955a 7.1GiB/27GiB(26%) CN:16 DL:65MiB ETA:5m15s]
[#c4955a 7.2GiB/27GiB(26%) CN:16 DL:65MiB ETA:5m14s]
```


[#c4955a 7.3GiB/27GiB(26%) CN:16 DL:69MiB ETA:4m56s]
[#c4955a 7.3GiB/27GiB(26%) CN:16 DL:69MiB ETA:4m55s]
[#c4955a 7.4GiB/27GiB(27%) CN:16 DL:69MiB ETA:4m54s]
[#c4955a 7.5GiB/27GiB(27%) CN:16 DL:68MiB ETA:4m59s]
[#c4955a 7.5GiB/27GiB(27%) CN:16 DL:67MiB ETA:4m59s]
[#c4955a 7.6GiB/27GiB(27%) CN:16 DL:64MiB ETA:5m14s]
[#c4955a 7.6GiB/27GiB(27%) CN:16 DL:66MiB ETA:5m5s]
[#c4955a 7.7GiB/27GiB(28%) CN:16 DL:66MiB ETA:5m4s]
[#c4955a 7.8GiB/27GiB(28%) CN:16 DL:66MiB ETA:5m]
[#c4955a 7.8GiB/27GiB(28%) CN:16 DL:66MiB ETA:5m1s]
[#c4955a 7.9GiB/27GiB(29%) CN:16 DL:68MiB ETA:4m52s]
[#c4955a 8.0GiB/27GiB(29%) CN:16 DL:68MiB ETA:4m48s]
[#c4955a 8.1GiB/27GiB(29%) CN:16 DL:69MiB ETA:4m46s]
[#c4955a 8.1GiB/27GiB(29%) CN:16 DL:70MiB ETA:4m40s]
[#c4955a 8.2GiB/27GiB(30%) CN:16 DL:70MiB ETA:4m37s]
[#c4955a 8.3GiB/27GiB(30%) CN:16 DL:73MiB ETA:4m25s]
[#c4955a 8.4GiB/27GiB(30%) CN:16 DL:73MiB ETA:4m25s]
[#c4955a 8.4GiB/27GiB(30%) CN:16 DL:72MiB ETA:4m27s]
[#c4955a 8.5GiB/27GiB(31%) CN:16 DL:72MiB ETA:4m25s]
[#c4955a 8.6GiB/27GiB(31%) CN:16 DL:72MiB ETA:4m25s]
[#c4955a 8.6GiB/27GiB(31%) CN:16 DL:72MiB ETA:4m25s]
[#c4955a 8.7GiB/27GiB(31%) CN:16 DL:71MiB ETA:4m27s]
[#c4955a 8.8GiB/27GiB(32%) CN:16 DL:71MiB ETA:4m25s]
[#c4955a 8.8GiB/27GiB(32%) CN:16 DL:71MiB ETA:4m27s]
[#c4955a 8.9GiB/27GiB(32%) CN:16 DL:70MiB ETA:4m28s]
[#c4955a 9.0GiB/27GiB(32%) CN:16 DL:71MiB ETA:4m23s]
[#c4955a 9.1GiB/27GiB(33%) CN:16 DL:71MiB ETA:4m22s]
[#c4955a 9.1GiB/27GiB(33%) CN:16 DL:71MiB ETA:4m23s]
[#c4955a 9.2GiB/27GiB(33%) CN:16 DL:69MiB ETA:4m26s]
[#c4955a 9.2GiB/27GiB(33%) CN:16 DL:70MiB ETA:4m24s]
[#c4955a 9.3GiB/27GiB(34%) CN:16 DL:70MiB ETA:4m23s]
[#c4955a 9.4GiB/27GiB(34%) CN:16 DL:70MiB ETA:4m20s]
[#c4955a 9.5GiB/27GiB(34%) CN:16 DL:70MiB ETA:4m21s]
[#c4955a 9.5GiB/27GiB(34%) CN:16 DL:69MiB ETA:4m21s]
[#c4955a 9.6GiB/27GiB(35%) CN:16 DL:69MiB ETA:4m20s]
[#c4955a 9.7GiB/27GiB(35%) CN:16 DL:70MiB ETA:4m16s]
[#c4955a 9.7GiB/27GiB(35%) CN:16 DL:71MiB ETA:4m14s]
[#c4955a 9.8GiB/27GiB(35%) CN:16 DL:72MiB ETA:4m7s]
[#c4955a 9.9GiB/27GiB(36%) CN:16 DL:73MiB ETA:4m4s]
[#c4955a 10GiB/27GiB(36%) CN:16 DL:74MiB ETA:4m]
[#c4955a 10GiB/27GiB(36%) CN:16 DL:74MiB ETA:3m59s]
[#c4955a 10GiB/27GiB(36%) CN:16 DL:74MiB ETA:3m57s]
[#c4955a 10GiB/27GiB(37%) CN:16 DL:75MiB ETA:3m54s]
[#c4955a 10GiB/27GiB(37%) CN:16 DL:76MiB ETA:3m50s]
[#c4955a 10GiB/27GiB(37%) CN:16 DL:76MiB ETA:3m50s]
[#c4955a 10GiB/27GiB(38%) CN:16 DL:73MiB ETA:3m55s]
[#c4955a 10GiB/27GiB(38%) CN:16 DL:73MiB ETA:3m57s]
[#c4955a 10GiB/27GiB(38%) CN:16 DL:72MiB ETA:3m57s]
[#c4955a 10GiB/27GiB(38%) CN:16 DL:73MiB ETA:3m55s]

[#c4955a 10GiB/27GiB(39%) CN:16 DL:72MiB ETA:3m54s]
[#c4955a 10GiB/27GiB(39%) CN:16 DL:72MiB ETA:3m54s]
[#c4955a 10GiB/27GiB(39%) CN:16 DL:72MiB ETA:3m55s]

*** Download Progress Summary as of Fri Oct 31 02:24:28 2025 ***

=====

[#c4955a 10GiB/27GiB(39%) CN:16 DL:72MiB ETA:3m54s]

FILE: /content/drive/MyDrive/eatsense/zips/eatsense_all_in_one.zip

[#c4955a 10GiB/27GiB(39%) CN:16 DL:72MiB ETA:3m54s]
[#c4955a 11GiB/27GiB(40%) CN:16 DL:72MiB ETA:3m53s]
[#c4955a 11GiB/27GiB(40%) CN:16 DL:73MiB ETA:3m48s]
[#c4955a 11GiB/27GiB(40%) CN:16 DL:75MiB ETA:3m42s]
[#c4955a 11GiB/27GiB(40%) CN:16 DL:75MiB ETA:3m41s]
[#c4955a 11GiB/27GiB(41%) CN:16 DL:73MiB ETA:3m44s]
[#c4955a 11GiB/27GiB(41%) CN:16 DL:72MiB ETA:3m47s]
[#c4955a 11GiB/27GiB(41%) CN:16 DL:71MiB ETA:3m51s]
[#c4955a 11GiB/27GiB(41%) CN:16 DL:69MiB ETA:3m55s]
[#c4955a 11GiB/27GiB(41%) CN:16 DL:67MiB ETA:4m]
[#c4955a 11GiB/27GiB(42%) CN:16 DL:66MiB ETA:4m4s]
[#c4955a 11GiB/27GiB(42%) CN:16 DL:65MiB ETA:4m7s]
[#c4955a 11GiB/27GiB(42%) CN:16 DL:65MiB ETA:4m7s]
[#c4955a 11GiB/27GiB(42%) CN:16 DL:64MiB ETA:4m6s]
[#c4955a 11GiB/27GiB(43%) CN:16 DL:65MiB ETA:4m3s]
[#c4955a 11GiB/27GiB(43%) CN:16 DL:66MiB ETA:3m58s]
[#c4955a 12GiB/27GiB(43%) CN:16 DL:67MiB ETA:3m52s]
[#c4955a 12GiB/27GiB(44%) CN:16 DL:70MiB ETA:3m44s]
[#c4955a 12GiB/27GiB(44%) CN:16 DL:71MiB ETA:3m40s]
[#c4955a 12GiB/27GiB(44%) CN:16 DL:72MiB ETA:3m36s]
[#c4955a 12GiB/27GiB(44%) CN:16 DL:72MiB ETA:3m33s]
[#c4955a 12GiB/27GiB(45%) CN:16 DL:73MiB ETA:3m31s]
[#c4955a 12GiB/27GiB(45%) CN:16 DL:72MiB ETA:3m31s]
[#c4955a 12GiB/27GiB(45%) CN:16 DL:72MiB ETA:3m32s]
[#c4955a 12GiB/27GiB(45%) CN:16 DL:72MiB ETA:3m29s]
[#c4955a 12GiB/27GiB(46%) CN:16 DL:73MiB ETA:3m27s]
[#c4955a 12GiB/27GiB(46%) CN:16 DL:72MiB ETA:3m26s]
[#c4955a 12GiB/27GiB(46%) CN:16 DL:73MiB ETA:3m24s]
[#c4955a 12GiB/27GiB(46%) CN:16 DL:73MiB ETA:3m23s]
[#c4955a 12GiB/27GiB(47%) CN:16 DL:73MiB ETA:3m23s]
[#c4955a 13GiB/27GiB(47%) CN:16 DL:73MiB ETA:3m21s]
[#c4955a 13GiB/27GiB(47%) CN:16 DL:73MiB ETA:3m19s]
[#c4955a 13GiB/27GiB(47%) CN:16 DL:72MiB ETA:3m23s]
[#c4955a 13GiB/27GiB(48%) CN:16 DL:72MiB ETA:3m22s]
[#c4955a 13GiB/27GiB(48%) CN:16 DL:72MiB ETA:3m21s]
[#c4955a 13GiB/27GiB(48%) CN:16 DL:71MiB ETA:3m20s]
[#c4955a 13GiB/27GiB(48%) CN:16 DL:71MiB ETA:3m19s]
[#c4955a 13GiB/27GiB(49%) CN:16 DL:71MiB ETA:3m19s]

```
[#c4955a 13GiB/27GiB(49%) CN:16 DL:71MiB ETA:3m18s]
[#c4955a 13GiB/27GiB(49%) CN:16 DL:71MiB ETA:3m18s]
[#c4955a 13GiB/27GiB(49%) CN:16 DL:71MiB ETA:3m16s]
[#c4955a 13GiB/27GiB(50%) CN:16 DL:71MiB ETA:3m14s]
[#c4955a 13GiB/27GiB(50%) CN:16 DL:73MiB ETA:3m10s]
[#c4955a 13GiB/27GiB(50%) CN:16 DL:71MiB ETA:3m13s]
[#c4955a 13GiB/27GiB(50%) CN:16 DL:72MiB ETA:3m10s]
[#c4955a 14GiB/27GiB(51%) CN:16 DL:71MiB ETA:3m11s]
[#c4955a 14GiB/27GiB(51%) CN:16 DL:71MiB ETA:3m10s]
[#c4955a 14GiB/27GiB(51%) CN:16 DL:71MiB ETA:3m9s]
[#c4955a 14GiB/27GiB(51%) CN:16 DL:72MiB ETA:3m6s]
[#c4955a 14GiB/27GiB(52%) CN:16 DL:72MiB ETA:3m4s]
[#c4955a 14GiB/27GiB(52%) CN:16 DL:72MiB ETA:3m3s]
[#c4955a 14GiB/27GiB(52%) CN:16 DL:72MiB ETA:3m2s]
[#c4955a 14GiB/27GiB(53%) CN:16 DL:73MiB ETA:2m58s]
[#c4955a 14GiB/27GiB(53%) CN:16 DL:74MiB ETA:2m55s]
[#c4955a 14GiB/27GiB(53%) CN:16 DL:73MiB ETA:2m58s]
[#c4955a 14GiB/27GiB(53%) CN:16 DL:73MiB ETA:2m57s]
[#c4955a 14GiB/27GiB(54%) CN:16 DL:73MiB ETA:2m56s]
[#c4955a 14GiB/27GiB(54%) CN:16 DL:71MiB ETA:3m1s]
[#c4955a 14GiB/27GiB(54%) CN:16 DL:70MiB ETA:3m]
[#c4955a 15GiB/27GiB(54%) CN:16 DL:70MiB ETA:3m1s]
[#c4955a 15GiB/27GiB(54%) CN:16 DL:69MiB ETA:3m1s]
```

*** Download Progress Summary as of Fri Oct 31 02:25:29 2025 ***

```
=====
=====
[#c4955a 15GiB/27GiB(55%) CN:16 DL:69MiB ETA:3m]
FILE: /content/drive/MyDrive/eatsense/zips/eatsense_all_in_one.zip
-----
```

```
-----
[#c4955a 15GiB/27GiB(55%) CN:16 DL:69MiB ETA:3m]
[#c4955a 15GiB/27GiB(55%) CN:16 DL:69MiB ETA:3m]
[#c4955a 15GiB/27GiB(55%) CN:16 DL:69MiB ETA:2m58s]
[#c4955a 15GiB/27GiB(56%) CN:16 DL:71MiB ETA:2m53s]
[#c4955a 15GiB/27GiB(56%) CN:16 DL:72MiB ETA:2m50s]
[#c4955a 15GiB/27GiB(56%) CN:16 DL:72MiB ETA:2m48s]
[#c4955a 15GiB/27GiB(56%) CN:16 DL:72MiB ETA:2m46s]
[#c4955a 15GiB/27GiB(57%) CN:16 DL:71MiB ETA:2m48s]
[#c4955a 15GiB/27GiB(57%) CN:16 DL:72MiB ETA:2m45s]
[#c4955a 15GiB/27GiB(57%) CN:16 DL:72MiB ETA:2m44s]
[#c4955a 15GiB/27GiB(57%) CN:16 DL:72MiB ETA:2m44s]
[#c4955a 15GiB/27GiB(58%) CN:16 DL:71MiB ETA:2m45s]
[#c4955a 16GiB/27GiB(58%) CN:16 DL:71MiB ETA:2m44s]
[#c4955a 16GiB/27GiB(58%) CN:16 DL:71MiB ETA:2m43s]
[#c4955a 16GiB/27GiB(58%) CN:16 DL:70MiB ETA:2m43s]
[#c4955a 16GiB/27GiB(59%) CN:16 DL:71MiB ETA:2m41s]
[#c4955a 16GiB/27GiB(59%) CN:16 DL:72MiB ETA:2m37s]
[#c4955a 16GiB/27GiB(59%) CN:16 DL:73MiB ETA:2m34s]
```

[#c4955a 16GiB/27GiB(59%) CN:16 DL:74MiB ETA:2m31s]
[#c4955a 16GiB/27GiB(60%) CN:16 DL:73MiB ETA:2m32s]
[#c4955a 16GiB/27GiB(60%) CN:16 DL:74MiB ETA:2m30s]
[#c4955a 16GiB/27GiB(60%) CN:16 DL:74MiB ETA:2m28s]
[#c4955a 16GiB/27GiB(60%) CN:16 DL:74MiB ETA:2m26s]
[#c4955a 16GiB/27GiB(61%) CN:16 DL:75MiB ETA:2m25s]
[#c4955a 16GiB/27GiB(61%) CN:16 DL:74MiB ETA:2m24s]
[#c4955a 16GiB/27GiB(61%) CN:16 DL:74MiB ETA:2m23s]
[#c4955a 17GiB/27GiB(61%) CN:16 DL:74MiB ETA:2m23s]
[#c4955a 17GiB/27GiB(62%) CN:16 DL:74MiB ETA:2m22s]
[#c4955a 17GiB/27GiB(62%) CN:16 DL:74MiB ETA:2m20s]
[#c4955a 17GiB/27GiB(62%) CN:16 DL:75MiB ETA:2m18s]
[#c4955a 17GiB/27GiB(63%) CN:16 DL:75MiB ETA:2m17s]
[#c4955a 17GiB/27GiB(63%) CN:16 DL:75MiB ETA:2m17s]
[#c4955a 17GiB/27GiB(63%) CN:16 DL:74MiB ETA:2m17s]
[#c4955a 17GiB/27GiB(63%) CN:16 DL:74MiB ETA:2m16s]
[#c4955a 17GiB/27GiB(64%) CN:16 DL:74MiB ETA:2m15s]
[#c4955a 17GiB/27GiB(64%) CN:16 DL:74MiB ETA:2m14s]
[#c4955a 17GiB/27GiB(64%) CN:16 DL:73MiB ETA:2m14s]
[#c4955a 17GiB/27GiB(64%) CN:16 DL:74MiB ETA:2m13s]
[#c4955a 17GiB/27GiB(65%) CN:16 DL:73MiB ETA:2m14s]
[#c4955a 17GiB/27GiB(65%) CN:16 DL:73MiB ETA:2m12s]
[#c4955a 18GiB/27GiB(65%) CN:16 DL:73MiB ETA:2m12s]
[#c4955a 18GiB/27GiB(65%) CN:16 DL:73MiB ETA:2m10s]
[#c4955a 18GiB/27GiB(66%) CN:16 DL:73MiB ETA:2m10s]
[#c4955a 18GiB/27GiB(66%) CN:16 DL:72MiB ETA:2m9s]
[#c4955a 18GiB/27GiB(66%) CN:16 DL:72MiB ETA:2m9s]
[#c4955a 18GiB/27GiB(66%) CN:16 DL:72MiB ETA:2m8s]
[#c4955a 18GiB/27GiB(67%) CN:16 DL:72MiB ETA:2m6s]
[#c4955a 18GiB/27GiB(67%) CN:16 DL:74MiB ETA:2m3s]
[#c4955a 18GiB/27GiB(67%) CN:16 DL:74MiB ETA:2m2s]
[#c4955a 18GiB/27GiB(68%) CN:16 DL:74MiB ETA:2m1s]
[#c4955a 18GiB/27GiB(68%) CN:16 DL:74MiB ETA:2m]
[#c4955a 18GiB/27GiB(68%) CN:16 DL:74MiB ETA:1m58s]
[#c4955a 18GiB/27GiB(68%) CN:16 DL:75MiB ETA:1m56s]
[#c4955a 18GiB/27GiB(69%) CN:16 DL:75MiB ETA:1m54s]
[#c4955a 19GiB/27GiB(69%) CN:16 DL:75MiB ETA:1m54s]
[#c4955a 19GiB/27GiB(69%) CN:16 DL:74MiB ETA:1m54s]
[#c4955a 19GiB/27GiB(69%) CN:16 DL:73MiB ETA:1m55s]
[#c4955a 19GiB/27GiB(70%) CN:16 DL:73MiB ETA:1m55s]
[#c4955a 19GiB/27GiB(70%) CN:16 DL:72MiB ETA:1m54s]
[#c4955a 19GiB/27GiB(70%) CN:16 DL:72MiB ETA:1m54s]
[#c4955a 19GiB/27GiB(70%) CN:16 DL:72MiB ETA:1m53s]

*** Download Progress Summary as of Fri Oct 31 02:26:29 2025 ***

=====

=====

[#c4955a 19GiB/27GiB(71%) CN:16 DL:72MiB ETA:1m52s]
FILE: /content/drive/MyDrive/eatsense/zips/eatsense_all_in_one.zip

```
[#c4955a 19GiB/27GiB(71%) CN:16 DL:72MiB ETA:1m52s]
[#c4955a 19GiB/27GiB(71%) CN:16 DL:72MiB ETA:1m51s]
[#c4955a 19GiB/27GiB(71%) CN:16 DL:71MiB ETA:1m51s]
[#c4955a 19GiB/27GiB(71%) CN:16 DL:71MiB ETA:1m50s]
[#c4955a 19GiB/27GiB(72%) CN:16 DL:72MiB ETA:1m47s]
[#c4955a 19GiB/27GiB(72%) CN:16 DL:73MiB ETA:1m45s]
[#c4955a 19GiB/27GiB(72%) CN:16 DL:73MiB ETA:1m44s]
[#c4955a 20GiB/27GiB(72%) CN:16 DL:72MiB ETA:1m45s]
[#c4955a 20GiB/27GiB(73%) CN:16 DL:70MiB ETA:1m46s]
[#c4955a 20GiB/27GiB(73%) CN:16 DL:70MiB ETA:1m46s]
[#c4955a 20GiB/27GiB(73%) CN:16 DL:68MiB ETA:1m47s]
[#c4955a 20GiB/27GiB(73%) CN:16 DL:68MiB ETA:1m48s]
[#c4955a 20GiB/27GiB(74%) CN:16 DL:67MiB ETA:1m48s]
[#c4955a 20GiB/27GiB(74%) CN:16 DL:68MiB ETA:1m46s]
[#c4955a 20GiB/27GiB(74%) CN:16 DL:67MiB ETA:1m45s]
[#c4955a 20GiB/27GiB(74%) CN:16 DL:67MiB ETA:1m45s]
[#c4955a 20GiB/27GiB(75%) CN:16 DL:67MiB ETA:1m43s]
[#c4955a 20GiB/27GiB(75%) CN:16 DL:68MiB ETA:1m40s]
[#c4955a 20GiB/27GiB(75%) CN:16 DL:69MiB ETA:1m38s]
[#c4955a 20GiB/27GiB(75%) CN:16 DL:69MiB ETA:1m37s]
[#c4955a 20GiB/27GiB(76%) CN:16 DL:71MiB ETA:1m34s]
[#c4955a 20GiB/27GiB(76%) CN:16 DL:71MiB ETA:1m32s]
[#c4955a 21GiB/27GiB(76%) CN:16 DL:72MiB ETA:1m30s]
[#c4955a 21GiB/27GiB(76%) CN:16 DL:73MiB ETA:1m28s]
[#c4955a 21GiB/27GiB(77%) CN:16 DL:73MiB ETA:1m27s]
[#c4955a 21GiB/27GiB(77%) CN:16 DL:73MiB ETA:1m26s]
[#c4955a 21GiB/27GiB(77%) CN:16 DL:73MiB ETA:1m25s]
[#c4955a 21GiB/27GiB(77%) CN:16 DL:73MiB ETA:1m25s]
[#c4955a 21GiB/27GiB(78%) CN:16 DL:73MiB ETA:1m24s]
[#c4955a 21GiB/27GiB(78%) CN:16 DL:72MiB ETA:1m23s]
[#c4955a 21GiB/27GiB(78%) CN:16 DL:72MiB ETA:1m23s]
[#c4955a 21GiB/27GiB(78%) CN:16 DL:71MiB ETA:1m23s]
[#c4955a 21GiB/27GiB(79%) CN:16 DL:70MiB ETA:1m23s]
[#c4955a 21GiB/27GiB(79%) CN:16 DL:70MiB ETA:1m22s]
[#c4955a 21GiB/27GiB(79%) CN:16 DL:69MiB ETA:1m22s]
[#c4955a 21GiB/27GiB(79%) CN:16 DL:70MiB ETA:1m20s]
[#c4955a 22GiB/27GiB(80%) CN:16 DL:70MiB ETA:1m19s]
[#c4955a 22GiB/27GiB(80%) CN:16 DL:71MiB ETA:1m16s]
[#c4955a 22GiB/27GiB(80%) CN:16 DL:72MiB ETA:1m14s]
[#c4955a 22GiB/27GiB(80%) CN:16 DL:73MiB ETA:1m13s]
[#c4955a 22GiB/27GiB(81%) CN:16 DL:74MiB ETA:1m10s]
[#c4955a 22GiB/27GiB(81%) CN:16 DL:74MiB ETA:1m9s]
[#c4955a 22GiB/27GiB(81%) CN:16 DL:76MiB ETA:1m7s]
[#c4955a 22GiB/27GiB(82%) CN:16 DL:75MiB ETA:1m6s]
[#c4955a 22GiB/27GiB(82%) CN:16 DL:75MiB ETA:1m5s]
[#c4955a 22GiB/27GiB(82%) CN:16 DL:75MiB ETA:1m4s]
[#c4955a 22GiB/27GiB(82%) CN:16 DL:76MiB ETA:1m3s]
```

```
[#c4955a 22GiB/27GiB(83%) CN:16 DL:76MiB ETA:1m2s]
[#c4955a 22GiB/27GiB(83%) CN:16 DL:75MiB ETA:1m1s]
[#c4955a 22GiB/27GiB(83%) CN:16 DL:75MiB ETA:1m]
[#c4955a 23GiB/27GiB(83%) CN:16 DL:76MiB ETA:59s]
[#c4955a 23GiB/27GiB(84%) CN:16 DL:76MiB ETA:58s]
[#c4955a 23GiB/27GiB(84%) CN:16 DL:75MiB ETA:57s]
[#c4955a 23GiB/27GiB(84%) CN:16 DL:75MiB ETA:56s]
[#c4955a 23GiB/27GiB(85%) CN:16 DL:76MiB ETA:55s]
[#c4955a 23GiB/27GiB(85%) CN:16 DL:76MiB ETA:54s]
[#c4955a 23GiB/27GiB(85%) CN:16 DL:74MiB ETA:54s]
[#c4955a 23GiB/27GiB(85%) CN:16 DL:74MiB ETA:53s]
[#c4955a 23GiB/27GiB(86%) CN:16 DL:73MiB ETA:53s]
[#c4955a 23GiB/27GiB(86%) CN:16 DL:73MiB ETA:52s]
[#c4955a 23GiB/27GiB(86%) CN:16 DL:73MiB ETA:51s]
```

*** Download Progress Summary as of Fri Oct 31 02:27:30 2025 ***

=====

```
=====
[#c4955a 23GiB/27GiB(86%) CN:16 DL:73MiB ETA:50s]
FILE: /content/drive/MyDrive/eatsense/zips/eatsense_all_in_one.zip
-----
```

```
[#c4955a 23GiB/27GiB(86%) CN:16 DL:73MiB ETA:50s]
[#c4955a 23GiB/27GiB(87%) CN:16 DL:73MiB ETA:49s]
[#c4955a 23GiB/27GiB(87%) CN:16 DL:73MiB ETA:48s]
[#c4955a 24GiB/27GiB(87%) CN:16 DL:70MiB ETA:49s]
[#c4955a 24GiB/27GiB(87%) CN:16 DL:67MiB ETA:51s]
[#c4955a 24GiB/27GiB(87%) CN:16 DL:68MiB ETA:49s]
[#c4955a 24GiB/27GiB(88%) CN:16 DL:68MiB ETA:48s]
[#c4955a 24GiB/27GiB(88%) CN:16 DL:68MiB ETA:47s]
[#c4955a 24GiB/27GiB(88%) CN:16 DL:67MiB ETA:46s]
[#c4955a 24GiB/27GiB(88%) CN:16 DL:67MiB ETA:45s]
[#c4955a 24GiB/27GiB(89%) CN:16 DL:68MiB ETA:44s]
[#c4955a 24GiB/27GiB(89%) CN:16 DL:67MiB ETA:43s]
[#c4955a 24GiB/27GiB(89%) CN:16 DL:67MiB ETA:42s]
[#c4955a 24GiB/27GiB(90%) CN:16 DL:73MiB ETA:38s]
[#c4955a 24GiB/27GiB(90%) CN:16 DL:73MiB ETA:36s]
[#c4955a 24GiB/27GiB(90%) CN:16 DL:74MiB ETA:35s]
[#c4955a 24GiB/27GiB(90%) CN:16 DL:74MiB ETA:34s]
[#c4955a 25GiB/27GiB(91%) CN:16 DL:74MiB ETA:33s]
[#c4955a 25GiB/27GiB(91%) CN:16 DL:74MiB ETA:32s]
[#c4955a 25GiB/27GiB(91%) CN:16 DL:74MiB ETA:31s]
[#c4955a 25GiB/27GiB(91%) CN:16 DL:73MiB ETA:30s]
[#c4955a 25GiB/27GiB(92%) CN:16 DL:72MiB ETA:30s]
[#c4955a 25GiB/27GiB(92%) CN:16 DL:73MiB ETA:29s]
[#c4955a 25GiB/27GiB(92%) CN:16 DL:72MiB ETA:28s]
[#c4955a 25GiB/27GiB(92%) CN:16 DL:73MiB ETA:27s]
[#c4955a 25GiB/27GiB(93%) CN:16 DL:73MiB ETA:26s]
[#c4955a 25GiB/27GiB(93%) CN:16 DL:73MiB ETA:25s]
[#c4955a 25GiB/27GiB(93%) CN:16 DL:73MiB ETA:24s]
```

```
[#c4955a 25GiB/27GiB(93%) CN:16 DL:73MiB ETA:23s]
[#c4955a 25GiB/27GiB(94%) CN:16 DL:74MiB ETA:21s]
[#c4955a 25GiB/27GiB(94%) CN:16 DL:74MiB ETA:20s]
[#c4955a 26GiB/27GiB(94%) CN:16 DL:75MiB ETA:19s]
[#c4955a 26GiB/27GiB(95%) CN:16 DL:74MiB ETA:18s]
[#c4955a 26GiB/27GiB(95%) CN:16 DL:74MiB ETA:17s]
[#c4955a 26GiB/27GiB(95%) CN:16 DL:75MiB ETA:16s]
[#c4955a 26GiB/27GiB(95%) CN:16 DL:74MiB ETA:15s]
[#c4955a 26GiB/27GiB(96%) CN:16 DL:74MiB ETA:14s]
[#c4955a 26GiB/27GiB(96%) CN:16 DL:74MiB ETA:13s]
[#c4955a 26GiB/27GiB(96%) CN:16 DL:75MiB ETA:12s]
[#c4955a 26GiB/27GiB(96%) CN:16 DL:75MiB ETA:11s]
[#c4955a 26GiB/27GiB(97%) CN:16 DL:75MiB ETA:10s]
[#c4955a 26GiB/27GiB(97%) CN:16 DL:75MiB ETA:9s]
[#c4955a 26GiB/27GiB(97%) CN:16 DL:75MiB ETA:8s]
[#c4955a 26GiB/27GiB(97%) CN:16 DL:75MiB ETA:7s]
[#c4955a 26GiB/27GiB(98%) CN:16 DL:75MiB ETA:6s]
[#c4955a 27GiB/27GiB(98%) CN:16 DL:74MiB ETA:5s]
[#c4955a 27GiB/27GiB(98%) CN:16 DL:74MiB ETA:4s]
[#c4955a 27GiB/27GiB(98%) CN:16 DL:74MiB ETA:3s]
[#c4955a 27GiB/27GiB(99%) CN:12 DL:73MiB ETA:2s]
[#c4955a 27GiB/27GiB(99%) CN:8 DL:72MiB ETA:1s]
[#c4955a 27GiB/27GiB(99%) CN:7 DL:68MiB ETA:1s]
[#c4955a 27GiB/27GiB(99%) CN:7 DL:64MiB ETA:1s]
[#c4955a 27GiB/27GiB(99%) CN:2 DL:60MiB]
[#c4955a 27GiB/27GiB(99%) CN:1 DL:54MiB]
[#c4955a 27GiB/27GiB(99%) CN:1 DL:49MiB]
[#c4955a 27GiB/27GiB(99%) CN:1 DL:42MiB]
[#c4955a 27GiB/27GiB(99%) CN:1 DL:35MiB]
[#c4955a 27GiB/27GiB(99%) CN:1 DL:28MiB]
[#c4955a 27GiB/27GiB(99%) CN:1 DL:21MiB]
```

10/31 02:28:28 [NOTICE] Download complete:
/content/drive/MyDrive/eatsense/zips/eatsense_all_in_one.zip

Download Results:

gid	stat	avg speed	path/URI
-----	------	-----------	----------

=====+=====+=====+=====
=====

c4955a OK	
70MiB/s	/content/drive/MyDrive/eatsense/zips/eatsense_all_in_one.zip

Status Legend:

(OK):download completed.

%bash

Inspect if the file is present and it's size:

ls -lh "\$ZIPS/eatsense_all_in_one.zip"

```
-rw----- 1 root root 28G Oct 31 02:28
/content/drive/MyDrive/eatsense/zips/eatsense_all_in_one.zip
```

Extract all archives

Here let's handle both the main bundle and any inner zip files

```
%%bash
set -e

ROOT="/content/drive/MyDrive/eatsense"
ZIPS="$ROOT/zips"
OUT="$ROOT"
TMP="/content/eatsense_tmp"

mkdir -p "$ZIPS"
"$OUT"/{rgb,depth,poses_true,poses_fake,test,all_misc} "$TMP"
shopt -s nullglob nocaseglob

# Unpack any master bundle that starts with DS or ds:
for z in "$ZIPS"/DS_*.zip "$ZIPS"/ds_*.zip; do
    [ -e "$z" ] || continue
    echo "Unpacking master bundle:" "$(basename "$z")"
    rm -rf "$TMP"/* 2>/dev/null || true
    unzip -n -q -DD "$z" -d "$TMP"

    # Move every inner zip found anywhere under TMP into ZIPS:
    while IFS= read -r inner; do
        echo "Staging inner zip:" "$(basename "$inner")"
        mv -n "$inner" "$ZIPS/"
    done <<(find "$TMP" -type f -iname '*.zip')

    # Copy non zip docs into all_misc:
    rsync -a --ignore-existing --exclude='*.zip' "$TMP"/
"$OUT/all_misc"/
done

# Extract all staged archives to organized folders:
for z in "$ZIPS"/*.zip; do
    base="$(basename "$z" | tr '[:upper:]' '[:lower:]')"
    if [[ "$base" == *"deepfaked"* || "$base" == *"rgb"* ]]; then
        out="$OUT/rgb"
    elif [[ "$base" == *"depth"* ]]; then
        out="$OUT/depth"
    elif [[ "$base" == *"true"* && "$base" == *"2d3d"* ]]; then
        out="$OUT/poses_true"
    elif [[ "$base" == *"faked"* && "$base" == *"2d3d"* ]]; then
        out="$OUT/poses_fake"
    elif [[ "$base" == *"test"* ]]; then
        out="$OUT/test"
    fi
    unzip -n -q -DD "$z" -d "$out"
done
```



```

out="$OUT/test"
else out="$OUT/all_misc"
fi
echo "Extracting:" "${basename "$z")" "->" "$out"
unzip -n -q -DD "$z" -d "$out"
done

echo "Extraction complete."

Extracting: eatsense_all_in_one.zip ->
/content/drive/MyDrive/eatsense/all_misc
Extraction complete.

%%bash
OUT="/content/drive/MyDrive/eatsense"

echo "Top-level sizes:"
du -h --max-depth=1 "$OUT" | sort -h

echo
echo "Counts by folder:"
for d in "$OUT"/*/*; do
    dcount=$(find "$d" -type d | wc -l)
    fcount=$(find "$d" -type f | wc -l)
    printf "%-30s dirs:%6d files:%7d\n" "${basename "$d")" "$dcount"
"$fcount"
done | sort

Top-level sizes:
4.0K /content/drive/MyDrive/eatsense/depth
4.0K /content/drive/MyDrive/eatsense/poses_fake
4.0K /content/drive/MyDrive/eatsense/poses_true
4.0K /content/drive/MyDrive/eatsense/rgb
4.0K /content/drive/MyDrive/eatsense/test
28G /content/drive/MyDrive/eatsense/all_misc
28G /content/drive/MyDrive/eatsense/zips
55G /content/drive/MyDrive/eatsense

Counts by folder:
all_misc          dirs:      1 files:      38
depth            dirs:      1 files:       0
poses_fake        dirs:      1 files:       0
poses_true        dirs:      1 files:       0
rgb               dirs:      1 files:       0
test              dirs:      1 files:       0
zips              dirs:      1 files:       1

```

Now let's check space and current layout:

```

%%bash
OUT="/content/drive/MyDrive/eatsense"
echo "Top level sizes:"
du -h --max-depth=1 "$OUT" | sort -h
echo
echo "Zip files:"
ls -lh "$OUT"/zips/*.zip 2>/dev/null || echo "No zips found"

```

```

Top level sizes:
4.0K /content/drive/MyDrive/eatsense/depth
4.0K /content/drive/MyDrive/eatsense/poses_fake
4.0K /content/drive/MyDrive/eatsense/poses_true
4.0K /content/drive/MyDrive/eatsense/rgb
4.0K /content/drive/MyDrive/eatsense/test
28G /content/drive/MyDrive/eatsense/all_misc
28G /content/drive/MyDrive/eatsense/zips
55G /content/drive/MyDrive/eatsense

```

```

Zip files:
-rw----- 1 root root 28G Oct 31 02:28
/content/drive/MyDrive/eatsense/zips/eatsense_all_in_one.zip

```

Here let's move any inner zips that landed in all_misc:

```

%%bash
set -e
OUT="/content/drive/MyDrive/eatsense"
ZIPS="$OUT/zips"
MISC="$OUT/all_misc"

mkdir -p "$ZIPS"
shopt -s nullglob nocaseglob
# move any zip files found under all_misc into zips:
find "$MISC" -type f -iname '*.zip' -print -exec mv -n {} "$ZIPS"/ \;
echo "Inner zips staged to zips"
ls -lh "$ZIPS" || true

```

```

/content/drive/MyDrive/eatsense/all_misc/all_depth.zip
/content/drive/MyDrive/eatsense/all_misc/deepfaked.zip
/content/drive/MyDrive/eatsense/all_misc/Faked_2D3D.zip
/content/drive/MyDrive/eatsense/all_misc/testset.zip
/content/drive/MyDrive/eatsense/all_misc/True2D3D.zip
Inner zips staged to zips
total 54G
-rw----- 1 root root 15G Oct 31 02:48 all_depth.zip
-rw----- 1 root root 11G Oct 31 02:51 deepfaked.zip
-rw----- 1 root root 28G Oct 31 02:28 eatsense_all_in_one.zip
-rw----- 1 root root 141M Oct 31 02:51 Faked_2D3D.zip
-rw----- 1 root root 410M Oct 31 02:51 testset.zip
-rw----- 1 root root 126M Oct 31 02:51 True2D3D.zip

```

```

%%bash
set -e
OUT="/content/drive/MyDrive/eatsense"
ZIPS="$OUT/zips"

# extract true poses only:
unzip -n -q -DD "$ZIPS/True2D3D.zip" -d "$OUT/poses_true"
rm -f "$ZIPS/True2D3D.zip"

# extract fake poses only:
unzip -n -q -DD "$ZIPS/Faked_2D3D.zip" -d "$OUT/poses_fake"
rm -f "$ZIPS/Faked_2D3D.zip"

# extract test set:
unzip -n -q -DD "$ZIPS/testset.zip" -d "$OUT/test"
rm -f "$ZIPS/testset.zip"

# extract rgb videos:
unzip -n -q -DD "$ZIPS/deepfaked.zip" -d "$OUT/rgb"
rm -f "$ZIPS/deepfaked.zip"

# extract depth maps:
unzip -n -q -DD "$ZIPS/all_depth.zip" -d "$OUT/depth"
rm -f "$ZIPS/all_depth.zip"

```

Sanity Checks

```

# Set paths once:
ROOT = Path("/content/drive/MyDrive/eatsense")
RGB_DIR      = ROOT / "rgb"
DEPTH_DIR    = ROOT / "depth" / "all_depth"
POSES_TRUE_DIR = ROOT / "poses_true"
POSES_FAKE_DIR = ROOT / "poses_fake"
TEST_DIR     = ROOT / "test"
MISC_DIR     = ROOT / "all_misc"
ZIPS_DIR     = ROOT / "zips"

paths = {
    "rgb": RGB_DIR,
    "depth": DEPTH_DIR,
    "poses_true": POSES_TRUE_DIR,
    "poses_fake": POSES_FAKE_DIR,
    "test": TEST_DIR,
    "all_misc": MISC_DIR,
    "zips": ZIPS_DIR,
}

for name, p in paths.items():
    print(name, "exists:", p.exists(), "path:", p)

```

```
rgb exists: True path: /content/drive/MyDrive/eatsense/rgb
depth exists: True path:
/content/drive/MyDrive/eatsense/depth/all_depth
poses_true exists: True path:
/content/drive/MyDrive/eatsense/poses_true
poses_fake exists: True path:
/content/drive/MyDrive/eatsense/poses_fake
test exists: True path: /content/drive/MyDrive/eatsense/test
all_misc exists: True path: /content/drive/MyDrive/eatsense/all_misc
zips exists: True path: /content/drive/MyDrive/eatsense/zips
```

```
%%bash
```

```
OUT="/content/drive/MyDrive/eatsense"
```

```
echo "Top-level sizes:"
```

```
du -h --max-depth=1 "$OUT" | sort -h
```

```
echo
```

```
echo "Counts by folder:"
```

```
for d in "$OUT"/*/*; do
```

```
    dcount=$(find "$d" -type d | wc -l)
```

```
    fcount=$(find "$d" -type f | wc -l)
```

```
    printf "%-30s dirs:%6d files:%7d\n" "$(basename "$d")" "$dcount"
"$fcount"
```

```
done | sort
```

```
Top-level sizes:
```

```
429M /content/drive/MyDrive/eatsense/test
580M /content/drive/MyDrive/eatsense/poses_true
637M /content/drive/MyDrive/eatsense/poses_fake
1.3G /content/drive/MyDrive/eatsense/all_misc
11G /content/drive/MyDrive/eatsense/rgb
15G /content/drive/MyDrive/eatsense/depth
28G /content/drive/MyDrive/eatsense/zips
56G /content/drive/MyDrive/eatsense
```

```
Counts by folder:
```

all_misc	dirs:	1	files:	33
depth	dirs:	2	files:	133
poses_fake	dirs:	3	files:	270
poses_true	dirs:	3	files:	270
rgb	dirs:	2	files:	135
test	dirs:	8	files:	36
zips	dirs:	1	files:	1

```
# Count common file types in each folder:
```

```
def count_types(root):
```

```
    files = glob.glob(os.path.join(str(root), "**", "*.*"),
recursive=True)
```

```

    exts = collections.Counter([os.path.splitext(p)[1].lower() for p
in files])
    return len(files), exts

for name, p in paths.items():
    total, exts = count_types(p)
    print(name, "total files:", total, "by extension:", dict(exts))

rgb total files: 135 by extension: {'mp4': 135}
depth total files: 133 by extension: {'mp4': 133}
poses_true total files: 270 by extension: {'csv': 270}
poses_fake total files: 270 by extension: {'csv': 270}
test total files: 36 by extension: {'mp4': 12, 'csv': 24}
all_misc total files: 33 by extension: {'mp4': 10, 'csv': 20,
'.pdf': 2, '.txt': 1}
zips total files: 1 by extension: {'zip': 1}

assert RGB_DIR.exists(), "rgb folder is missing"
assert DEPTH_DIR.exists(), "depth folder is missing"
assert POSES_TRUE_DIR.exists(), "poses_true folder is missing"
assert POSES_FAKE_DIR.exists(), "poses_fake folder is missing"
assert TEST_DIR.exists(), "test folder is missing"

rgb_mp4 = glob.glob(os.path.join(str(RGB_DIR), "**", "*.mp4"),
recursive=True)
depth_mp4= glob.glob(os.path.join(str(DEPTH_DIR), "**", "*.mp4"),
recursive=True)
true_csv = glob.glob(os.path.join(str(POSES_TRUE_DIR), "**", "*.csv"),
recursive=True)
fake_csv = glob.glob(os.path.join(str(POSES_FAKE_DIR), "**", "*.csv"),
recursive=True)
test_any = glob.glob(os.path.join(str(TEST_DIR), "**", "*.*"),
recursive=True)

print("rgb videos:", len(rgb_mp4))
print("depth videos:", len(depth_mp4))
print("true pose csv:", len(true_csv))
print("fake pose csv:", len(fake_csv))
print("test files:", len(test_any))

assert len(rgb_mp4) > 0, "no rgb videos found"
assert len(depth_mp4)> 0, "no depth videos found"
assert len(true_csv) > 0, "no true pose csv found"
assert len(fake_csv) > 0, "no fake pose csv found"
assert len(test_any) > 0, "no test files found"

rgb videos: 135
depth videos: 133
true pose csv: 270

```

```

fake pose csv: 270
test files: 36

def sample(paths, k=3):
    return paths[:k] if len(paths) >= k else paths

print("rgb samples:", sample(sorted(rgb_mp4)))
print("depth samples:", sample(sorted(depth_mp4)))
print("true csv samples:", sample(sorted(true_csv)))
print("fake csv samples:", sample(sorted(fake_csv)))

rgb samples:
['/content/drive/MyDrive/eatsense/rgb/deepfaked/20210518_230219_anonym
ized.mp4',
'/content/drive/MyDrive/eatsense/rgb/deepfaked/20210523_202300_anonymi
zed.mp4',
'/content/drive/MyDrive/eatsense/rgb/deepfaked/20210529_150552_anonymi
zed.mp4']
depth samples:
['/content/drive/MyDrive/eatsense/depth/all_depth/20210518_230219_dept
h.mp4',
'/content/drive/MyDrive/eatsense/depth/all_depth/20210523_202300_depth
.mp4',
'/content/drive/MyDrive/eatsense/depth/all_depth/20210529_150552_depth
.mp4']
true csv samples:
['/content/drive/MyDrive/eatsense/poses_true/True2D/20210518_230219.mp
4_preds_HigherHRNet.csv',
'/content/drive/MyDrive/eatsense/poses_true/True2D/20210523_202300.mp4
_preds_HigherHRNet.csv',
'/content/drive/MyDrive/eatsense/poses_true/True2D/20210529_150552.mp4
_preds_HigherHRNet.csv']
fake csv samples:
['/content/drive/MyDrive/eatsense/poses_fake/Faked2D/20210518_230219_p
reds_HigherHRNet.csv',
'/content/drive/MyDrive/eatsense/poses_fake/Faked2D/20210523_202300_pr
eds_HigherHRNet.csv',
'/content/drive/MyDrive/eatsense/poses_fake/Faked2D/20210529_150552_pr
eds_HigherHRNet.csv']

rgb_stems = [os.path.splitext(os.path.basename(p))[0] for p in
rgb_mp4]
true_stems = [os.path.basename(p) for p in true_csv]
fake_stems = [os.path.basename(p) for p in fake_csv]

hits_true = sum(any(s in t for t in true_stems) for s in rgb_stems)
hits_fake = sum(any(s in t for t in fake_stems) for s in rgb_stems)

print("rgb count:", len(rgb_stems))

```

```

print("rgb that match some true csv:", hits_true)
print("rgb that match some fake csv:", hits_fake)

rgb count: 135
rgb that match some true csv: 0
rgb that match some fake csv: 0

%%bash
ROOT="/content/drive/MyDrive/eatsense"
echo "Zip files outside zips:"
find "$ROOT" -type f -iname '*.zip' ! -path "$ROOT/zips/*" -print ||
true

echo
echo "Nested zips under data folders:"
find "$ROOT"/{rgb,depth,poses_true,poses_fake,test} -type f -iname
'*.zip' -print || true

Zip files outside zips:

Nested zips under data folders:

ROOT = "/content/drive/MyDrive/eatsense"
RGB_DIR = f"{ROOT}/rgb"
POSES_TRUE_DIR = f"{ROOT}/poses_true"
POSES_FAKE_DIR = f"{ROOT}/poses_fake"

pat = re.compile(r"(\d{8}_\d{6})")

def key_from_path(p):
    n = os.path.basename(p).lower()
    m = pat.search(n)
    return m.group(1) if m else None

rgb = sorted(glob.glob(os.path.join(RGB_DIR, "**", "*.mp4"),
recursive=True))
true_csv = sorted(glob.glob(os.path.join(POSES_TRUE_DIR, "**",
"*.csv"), recursive=True))
fake_csv = sorted(glob.glob(os.path.join(POSES_FAKE_DIR, "**",
"*.csv"), recursive=True))

rgb_keys = {key_from_path(p): p for p in rgb if key_from_path(p)}
true_keys = {key_from_path(p): p for p in true_csv if
key_from_path(p)}
fake_keys = {key_from_path(p): p for p in fake_csv if
key_from_path(p)}

print("rgb keys:", len(rgb_keys))
print("true pose keys:", len(true_keys))
print("fake pose keys:", len(fake_keys))

```

```

overlap_true = sorted(set(rgb_keys) & set(true_keys))
overlap_fake = sorted(set(rgb_keys) & set(fake_keys))

print("rgb match with true pose by timestamp:", len(overlap_true))
print("rgb match with fake pose by timestamp:", len(overlap_fake))

rgb keys: 135
true pose keys: 135
fake pose keys: 135
rgb match with true pose by timestamp: 135
rgb match with fake pose by timestamp: 135

def sample_pairs(keys, left_map, right_map, k=5):
    out = []
    for i, kk in enumerate(keys[:k]):
        out.append((kk, left_map[kk], right_map[kk]))
    return out

print("example matches with true:")
for kk, rv, tv in sample_pairs(overlap_true, rgb_keys, true_keys):
    print(kk)
    print("  rgb :", rv)
    print("  true:", tv)

print("\nextample matches with fake:")
for kk, rv, fv in sample_pairs(overlap_fake, rgb_keys, fake_keys):
    print(kk)
    print("  rgb :", rv)
    print("  fake:", fv)

example matches with true:
20210518_230219
  rgb :
/content/drive/MyDrive/eatsense/rgb/deepfaked/20210518_230219_anonymiz
ed.mp4
  true:
/content/drive/MyDrive/eatsense/poses_true/True3D/20210518_230219.csv
20210523_202300
  rgb :
/content/drive/MyDrive/eatsense/rgb/deepfaked/20210523_202300_anonymiz
ed.mp4
  true:
/content/drive/MyDrive/eatsense/poses_true/True3D/20210523_202300.csv
20210529_150552
  rgb :
/content/drive/MyDrive/eatsense/rgb/deepfaked/20210529_150552_anonymiz
ed.mp4
  true:
/content/drive/MyDrive/eatsense/poses_true/True3D/20210529_150552.csv
20210529_153708

```



```

    rgb :
/content/drive/MyDrive/eatsense/rgb/deepfaked/20210529_153708_anonymiz
ed.mp4
    true:
/content/drive/MyDrive/eatsense/poses_true/True3D/20210529_153708.csv
20210530_153343
    rgb :
/content/drive/MyDrive/eatsense/rgb/deepfaked/20210530_153343_anonymiz
ed.mp4
    true:
/content/drive/MyDrive/eatsense/poses_true/True3D/20210530_153343.csv

example matches with fake:
20210518_230219
    rgb :
/content/drive/MyDrive/eatsense/rgb/deepfaked/20210518_230219_anonymiz
ed.mp4
    fake:
/content/drive/MyDrive/eatsense/poses_fake/Faked3D/20210518_230219.csv
20210523_202300
    rgb :
/content/drive/MyDrive/eatsense/rgb/deepfaked/20210523_202300_anonymiz
ed.mp4
    fake:
/content/drive/MyDrive/eatsense/poses_fake/Faked3D/20210523_202300.csv
20210529_150552
    rgb :
/content/drive/MyDrive/eatsense/rgb/deepfaked/20210529_150552_anonymiz
ed.mp4
    fake:
/content/drive/MyDrive/eatsense/poses_fake/Faked3D/20210529_150552.csv
20210529_153708
    rgb :
/content/drive/MyDrive/eatsense/rgb/deepfaked/20210529_153708_anonymiz
ed.mp4
    fake:
/content/drive/MyDrive/eatsense/poses_fake/Faked3D/20210529_153708.csv
20210530_153343
    rgb :
/content/drive/MyDrive/eatsense/rgb/deepfaked/20210530_153343_anonymiz
ed.mp4
    fake:
/content/drive/MyDrive/eatsense/poses_fake/Faked3D/20210530_153343.csv

# What did not match:
missing_true = sorted(set(rgb_keys) - set(true_keys))
missing_fake = sorted(set(rgb_keys) - set(fake_keys))
print("rgb without true poses:", len(missing_true))
print("rgb without fake poses:", len(missing_fake))

```

```
# peek a few to understand naming:
for kk in missing_true[:5]:
    print("no true for:", kk, "rgb path:", rgb_keys[kk])

rgb without true poses: 0
rgb without fake poses: 0
```

Build a pairing manifest

Here's let's create one table that pairs each sample across all folders using a single timestamp key `yyyyMMdd_HHmmss`. This manifest becomes the source of truth for loading aligned data.

This table will help us:

- Keeps rgb, depth, and pose files in sync.
- Makes data loading simple; I can iterate rows and know I have a complete set.
- Avoids silent errors: if any file is missing the row will not be created.
- Speeds up experiments: I can split, shuffle, and filter with one dataframe.
- Train or evaluate by iterating rows; each row is one synchronized sample.
- Build train or validation splits by sampling keys; store the split ids.

Table Contents:

- `key`: timestamp string used to join files.
- `rgb`: full path to the rgb video.
- `depth`: full path to the depth video.
- `poses underscore true`: full path to the true pose csv.
- `poses underscore fake`: full path to the fake pose csv.

```
# Alright now let's build a manifest keyed by timestamp:

ROOT = "/content/drive/MyDrive/eatsense"
RGB_DIR = f"{ROOT}/rgb"
DEPTH_DIR = f"{ROOT}/depth/all_depth"
TRUE_DIR = f"{ROOT}/poses_true"
FAKE_DIR = f"{ROOT}/poses_fake"

pat = re.compile(r"(\d{8}_\d{6})")

def key_from_path(p):
    m = pat.search(os.path.basename(p).lower())
    return m.group(1) if m else None

rgb = glob.glob(os.path.join(RGB_DIR, "**", "*.mp4"),
recursive=True)
depth = glob.glob(os.path.join(DEPTH_DIR, "**", "*.mp4"),
recursive=True)
true_ = glob.glob(os.path.join(TRUE_DIR, "**", "*.csv"),
recursive=True)
```

```

fake_ = glob.glob(os.path.join(FAKE_DIR,  "**", "*.csv"),
recursive=True)

rgb_map   = {key_from_path(p): p for p in rgb   if key_from_path(p)}
depth_map = {key_from_path(p): p for p in depth if key_from_path(p)}
true_map  = {key_from_path(p): p for p in true_ if key_from_path(p)}
fake_map  = {key_from_path(p): p for p in fake_ if key_from_path(p)}

keys = sorted(set(rgb_map) & set(depth_map) & set(true_map) &
set(fake_map))
print("paired keys count:", len(keys))

```

```

rows = []
for k in keys:
    rows.append(dict(
        key=k,
        rgb=rgb_map[k],
        depth=depth_map[k],
        poses_true=true_map[k],
        poses_fake=fake_map[k],
    ))

```

```

manifest =
pd.DataFrame(rows).sort_values("key").reset_index(drop=True)
print("manifest rows:", len(manifest))
manifest.head(10)

```

```

paired keys count: 133
manifest rows: 133

```

```

{"summary":{"\n  \"name\": \"manifest\", \n  \"rows\": 133, \n
\"fields\": [\n    {\n      \"column\": \"key\", \n
\"properties\": {\n        \"dtype\": \"string\", \n
\"num_unique_values\": 133, \n        \"samples\": [\n
\"20220811_122341\", \n          \"20221013_142709\", \n
\"20210810_163907\", \n        ], \n        \"semantic_type\": \"\", \n
\"description\": \"\" \n      } \n    }, \n    {\n      \"column\":
\"rgb\", \n      \"properties\": {\n        \"dtype\": \"string\", \n
\"num_unique_values\": 133, \n        \"samples\": [\n
\"/content/drive/MyDrive/eatsense/rgb/deepfaked/20220811_122341_anonym
ized.mp4\", \n
\"/content/drive/MyDrive/eatsense/rgb/deepfaked/out_rgb_[20221013_1427
09]_anonymized.mp4\", \n
\"/content/drive/MyDrive/eatsense/rgb/deepfaked/20210810_163907_anonym
ized.mp4\", \n        ], \n        \"semantic_type\": \"\", \n
\"description\": \"\" \n      } \n    }, \n    {\n      \"column\":
\"depth\", \n      \"properties\": {\n        \"dtype\": \"string\", \n
\"num_unique_values\": 133, \n        \"samples\": [\n
\"/content/drive/MyDrive/eatsense/depth/all_depth/20220811_122341_dept
h.mp4\", \n

```

```

\"/content/drive/MyDrive/eatsense/depth/all_depth/20221013_142709_dept
h.mp4\", \n
\"/content/drive/MyDrive/eatsense/depth/all_depth/20210810_163907_dept
h.mp4\" \n      ], \n      \"semantic_type\": \"\", \n
\"description\": \"\" \n      } \n      }, \n      { \n      \"column\":
\"poses_true\", \n      \"properties\": { \n      \"dtype\":
\"string\", \n      \"num_unique_values\": 133, \n      \"samples\":
[ \n
\"/content/drive/MyDrive/eatsense/poses_true/True3D/20220811_122341.cs
v\", \n
\"/content/drive/MyDrive/eatsense/poses_true/True3D/20221013_142709.cs
v\", \n
\"/content/drive/MyDrive/eatsense/poses_true/True3D/20210810_163907.cs
v\" \n      ], \n      \"semantic_type\": \"\", \n
\"description\": \"\" \n      } \n      }, \n      { \n      \"column\":
\"poses_fake\", \n      \"properties\": { \n      \"dtype\":
\"string\", \n      \"num_unique_values\": 133, \n      \"samples\":
[ \n
\"/content/drive/MyDrive/eatsense/poses_fake/Faked3D/20220811_122341.c
sv\", \n
\"/content/drive/MyDrive/eatsense/poses_fake/Faked3D/20221013_142709.c
sv\", \n
\"/content/drive/MyDrive/eatsense/poses_fake/Faked3D/20210810_163907.c
sv\" \n      ], \n      \"semantic_type\": \"\", \n
\"description\": \"\" \n      } \n      } \n      ] \n
n} \", \"type\": \"dataframe\", \"variable_name\": \"manifest\"}

```

Now let's save the manifest:

```

out_csv = f\"{ROOT}/manifest_eatsense.csv\"
manifest.to_csv(out_csv, index=False)
print(\"saved:\", out_csv)

```

saved: /content/drive/MyDrive/eatsense/manifest_eatsense.csv

Confirm that every file exists on disk:

```

missing = []
for col in [\"rgb\", \"depth\", \"poses_true\", \"poses_fake\"]:
    bad = manifest[~manifest[col].map(os.path.exists)]
    if len(bad):
        missing.append((col, len(bad)))
print(\"missing summary:\", missing if missing else \"none\")

```

Show a few joined examples:

```

for i in range(3):
    r = manifest.iloc[i]
    print(i, r.key)
    print(\"  rgb    \": r.rgb)
    print(\"  depth  \": r.depth)
    print(\"  true   \": r.poses_true)
    print(\"  fake   \": r.poses_fake)

```

```

missing summary: none
0 20210518_230219
  rgb :
/content/drive/MyDrive/eatsense/rgb/deepfaked/20210518_230219_anonymiz
ed.mp4
  depth :
/content/drive/MyDrive/eatsense/depth/all_depth/20210518_230219_depth.
mp4
  true :
/content/drive/MyDrive/eatsense/poses_true/True3D/20210518_230219.csv
  fake :
/content/drive/MyDrive/eatsense/poses_fake/Faked3D/20210518_230219.csv
1 20210523_202300
  rgb :
/content/drive/MyDrive/eatsense/rgb/deepfaked/20210523_202300_anonymiz
ed.mp4
  depth :
/content/drive/MyDrive/eatsense/depth/all_depth/20210523_202300_depth.
mp4
  true :
/content/drive/MyDrive/eatsense/poses_true/True3D/20210523_202300.csv
  fake :
/content/drive/MyDrive/eatsense/poses_fake/Faked3D/20210523_202300.csv
2 20210529_150552
  rgb :
/content/drive/MyDrive/eatsense/rgb/deepfaked/20210529_150552_anonymiz
ed.mp4
  depth :
/content/drive/MyDrive/eatsense/depth/all_depth/20210529_150552_depth.
mp4
  true :
/content/drive/MyDrive/eatsense/poses_true/True3D/20210529_150552.csv
  fake :
/content/drive/MyDrive/eatsense/poses_fake/Faked3D/20210529_150552.csv

# Here let's preview a synchronized triplet:
# Let's display one rgb frame and one depth frame together, then peek
at the first rows of both pose files:
r = manifest.sample(1, random_state=7).iloc[0]
print("key:", r.key)

# grab first frame from rgb and depth:
def first_frame(path):
    cap = cv2.VideoCapture(path)
    ok, frame = cap.read()
    cap.release()
    return frame if ok else None

rgb_frame = first_frame(r.rgb)
depth_frame = first_frame(r.depth)

```

```
print("rgb frame shape:", None if rgb_frame is None else
      rgb_frame.shape)
print("depth frame shape:", None if depth_frame is None else
      depth_frame.shape)

if rgb_frame is not None:
    cv2_imshow(rgb_frame)
if depth_frame is not None:
    cv2_imshow(depth_frame)

# peek at pose tables:
df_true = pd.read_csv(r.poses_true)
df_fake = pd.read_csv(r.poses_fake)
display(df_true.head(5))
display(df_fake.head(5))

key: 20221013_132443
rgb frame shape: (480, 640, 3)
depth frame shape: (480, 640, 3)
```





```
{"type": "dataframe"}
```

```
{"type": "dataframe"}
```

Final Checklist

```
%%bash
OUT="/content/drive/MyDrive/eatsense"
echo "final sizes"
du -h --max-depth=1 "$OUT" | sort -h

final sizes
429M /content/drive/MyDrive/eatsense/test
580M /content/drive/MyDrive/eatsense/poses_true
637M /content/drive/MyDrive/eatsense/poses_fake
1.3G /content/drive/MyDrive/eatsense/all_misc
11G /content/drive/MyDrive/eatsense/rgb
15G /content/drive/MyDrive/eatsense/depth
28G /content/drive/MyDrive/eatsense/zips
56G /content/drive/MyDrive/eatsense
```

- Top level sizes look correct.

- Folders exist and counts match expectations.
- Normalized timestamp keys align rgb with poses and depth.
- Manifest is saved to Drive.
- Cleanup: let's remove zips to save in the space:

Free space check:

```
!df -h /content/drive | tail -n 1
```

```
drive          100G   86G   15G   86% /content/drive
```

Remove all zipped files

```
%%bash
set -e
ROOT="/content/drive/MyDrive/eatsense"
ZIPS="$ROOT/zips"

echo "Before:"
du -h --max-depth=1 "$ROOT" | sort -h
echo

echo "Removing zips in zips folder;"
rm -f "$ZIPS"/*.zip 2>/dev/null || true

echo "Removing any nested zips under data folders;"
find "$ROOT"/{rgb,depth,poses_true,poses_fake,test,all_misc} -type f -
iname "*.zip" -print -delete 2>/dev/null || true

echo "Remove empty zips folder if empty;"
rmdir "$ZIPS" 2>/dev/null || true

echo
echo "After:"
du -h --max-depth=1 "$ROOT" | sort -h

Before:
429M /content/drive/MyDrive/eatsense/test
580M /content/drive/MyDrive/eatsense/poses_true
637M /content/drive/MyDrive/eatsense/poses_fake
1.3G /content/drive/MyDrive/eatsense/all_misc
11G /content/drive/MyDrive/eatsense/rgb
15G /content/drive/MyDrive/eatsense/depth
28G /content/drive/MyDrive/eatsense/zips
56G /content/drive/MyDrive/eatsense

Removing zips in zips folder;
Removing any nested zips under data folders;
Remove empty zips folder if empty;

After:
```



```
429M /content/drive/MyDrive/eatsense/test
580M /content/drive/MyDrive/eatsense/poses_true
637M /content/drive/MyDrive/eatsense/poses_fake
1.3G /content/drive/MyDrive/eatsense/all_misc
11G /content/drive/MyDrive/eatsense/rgb
15G /content/drive/MyDrive/eatsense/depth
29G /content/drive/MyDrive/eatsense
```

Free space check:

```
!df -h /content/drive | tail -n 1
```

```
drive          100G   33G   68G  33% /content/drive
```