FineTuneModNet

December 3, 2025

[21]:

```
!nvidia-smi
```

```
Thu Oct 16 15:47:47 2025
+-------------------------------------------------------------------------------
----------+
| NVIDIA-SMI 550.54.15              Driver Version: 550.54.15      CUDA Version:
12.4     |
|-----------------------------------------+------------------------+------------
----------+
| GPU  Name                 Persistence-M | Bus-Id          Disp.A | Volatile
Uncorr. ECC |
| Fan  Temp   Perf          Pwr:Usage/Cap |           Memory-Usage | GPU-Util
Compute M. |
|                                         |                        |
MIG M. |
|=========================================+========================+============
==========|
|   0  Tesla T4                       Off |   00000000:00:04.0 Off |
0 |
| N/A   39C    P8              9W /   70W |       2MiB /  15360MiB |      0%
Default |
|                                         |                        |
N/A |
+-----------------------------------------+------------------------+------------
----------+

+-------------------------------------------------------------------------------
----------+
| Processes:
|
|  GPU   GI   CI        PID   Type   Process name
GPU Memory |
|        ID   ID
Usage      |
|===============================================================================
==========|
|  No running processes found
|
+-------------------------------------------------------------------------------
----------+
```

[22]:

```
!pip -q install --upgrade pip
!pip -q install opencv-python pillow matplotlib albumentations tqdm einops

# (Colab ships with torch/torchvision; uncomment if you need a specific CUDA␣
 ↪build)
# !pip -q install torch torchvision --index-url https://download.pytorch.org/
 ↪whl/cu121

# Clone official MODNet
%cd /content
!git clone https://github.com/ZHKKKe/MODNet.git
%cd MODNet
```

```
                         0.0/1.8 MB
? eta -:--:--
                      1.0/1.8

MB 31.3 MB/s eta 0:00:01
                      1.8/1.8 MB 32.5

MB/s eta 0:00:00
/content
Cloning into 'MODNet'…
remote: Enumerating objects: 276, done.
remote: Counting objects: 100% (40/40), done.
remote: Compressing objects: 100% (24/24), done.
remote: Total 276 (delta 21), reused 16 (delta 16), pack-reused 236 (from 1)
Receiving objects: 100% (276/276), 60.77 MiB | 32.82 MiB/s, done.
Resolving deltas: 100% (100/100), done.
/content/MODNet
```

[1]:

```python
import os, sys, glob, math, random, time
from pathlib import Path
import numpy as np
import cv2
import torch
import torch.nn as nn
import torch.nn.functional as F
from torch.utils.data import Dataset, DataLoader
from torchvision import transforms
import albumentations as A
from albumentations.pytorch import ToTensorV2
import matplotlib.pyplot as plt
from tqdm.auto import tqdm
from PIL import Image
```

```python



device = 'cuda' if torch.cuda.is_available() else 'cpu'
device
```

[1]:

```
'cuda'
```

[2]:

```python
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

[49]:

```python
from pathlib import Path
import shutil
import random

# Define source dataset roots
AIM_ROOT = Path('/content/drive/MyDrive/AIM-500')
AIT_ROOT = Path('/content/drive/MyDrive/AIT_Faculty')

# Define destination root
DATA_ROOT = Path('/content/drive/MyDrive/modnet_data')

# Define split structure
splits = ['train', 'val', 'test']
subdirs = ['images', 'mattes']

# Create destination directories
for split in splits:
    for subdir in subdirs:
        (DATA_ROOT / split / subdir).mkdir(parents=True, exist_ok=True)
```

[39]:

```python
train_img_dir = DATA_ROOT / 'train' / 'images'
train_matte_dir = DATA_ROOT / 'train' / 'mattes'
val_img_dir = DATA_ROOT / 'val' / 'images'
val_matte_dir = DATA_ROOT / 'val' / 'mattes'
test_img_dir = DATA_ROOT / 'test' / 'images'
test_matte_dir = DATA_ROOT / 'test' / 'mattes'
```

[40]:

```python
print(train_img_dir)
print(train_matte_dir)
print(val_img_dir)
print(val_matte_dir)
```

```
/content/drive/MyDrive/modnet_data/train/images
/content/drive/MyDrive/modnet_data/train/mattes
/content/drive/MyDrive/modnet_data/val/images
```

```
/content/drive/MyDrive/modnet_data/val/mattes
```

[41]:

```python
# =====================================================
#    collect_pairs() - Match and auto-rename mattes
# =====================================================
from pathlib import Path
import os

def collect_pairs(original_dir: Path, mask_dir: Path, auto_rename=True):
    """
    Collect matching image-matte pairs even if:
      • File extensions differ (.jpg, .png, etc.)
      • Matte filenames include '_matte' suffix
    Optionally auto-renames mattes to match their images.
    Returns: list of (image_path, matte_path) tuples
    """
    original_files = list(original_dir.glob('*'))
    mask_files = list(mask_dir.glob('*'))

    # Build map of matte stems (handle '_matte' suffix)
    mask_map = {}
    for f in mask_files:
        stem = f.stem
        if stem.endswith('_matte'):
            stem = stem[:-6]  # remove suffix
        mask_map[stem] = f

    pairs = []
    renamed = 0
    for img_path in original_files:
        stem = img_path.stem

        # Direct match
        if stem in mask_map:
            matte_path = mask_map[stem]

            # Rename matte if name contains "_matte"
            if auto_rename and "_matte" in matte_path.stem:
                new_name = stem + matte_path.suffix
                new_path = matte_path.with_name(new_name)
                if not new_path.exists():
                    os.rename(matte_path, new_path)
                    matte_path = new_path
                    renamed += 1
                    print(f"  Renamed: {matte_path.name}")
                else:
                    print(f"  Skipped rename (exists): {new_name}")
            pairs.append((img_path, matte_path))

        else:
            # Try fuzzy match (partial overlap)
            match = next((m for k, m in mask_map.items() if stem in k or k in stem), None)
            if match:
                pairs.append((img_path, match))
            else:
                print(f"  No matching matte found for: {img_path.name}")

    print(f"\n  Collected {len(pairs)} valid pairs |   Renamed: {renamed}")
    return pairs
```

[55]:

```
# Function to copy image-mask pairs to a split folder, skipping if already copied
def copy_split(pairs, split, dest_root):
    i=0;
    for img_path, mask_path in pairs:
        dest_img = dest_root / split / 'images' / img_path.name
        dest_mask = dest_root / split / 'mattes' / mask_path.name

        if not dest_img.exists() and not dest_mask.exists():
            i+=1
            print(str(i)+f" copying {img_path.name}")
            shutil.copy(img_path, dest_img)
            shutil.copy(mask_path, dest_mask)
        # if not dest_mask.exists():
        #     shutil.copy(mask_path, dest_mask)
```

[51]:

```
# Function to split and copy a dataset directly
def split_and_copy(original_dir, mask_dir, split_ratios, dest_root):
    pairs = collect_pairs(original_dir, mask_dir)
    print(f"Found {len(pairs)} pairs in {original_dir}")

    random.shuffle(pairs)
    total = len(pairs)
    train_end = int(split_ratios[0] * total)
    val_end = train_end + int(split_ratios[1] * total)

    splits_map = {
        'train': pairs[:train_end],
        'val': pairs[train_end:val_end],
        'test': pairs[val_end:]
    }
    for split, items in splits_map.items():
        copy_split(items, split, dest_root)
```

[52]:
```
# Apply to AIM and AIT datasets
split_ratios = [0.7, 0.15, 0.15]  # train, val, test
```

[53]:
```
print("AIM images:", len(list(Path(AIM_ROOT/'original').glob('*'))))
print("AIM mattes:", len(list(Path(AIM_ROOT / 'mask').glob('*'))))
print("AIT images:", len(list(Path(AIT_ROOT / 'images').glob('*'))))
print("AIT mattes:", len(list(Path(AIT_ROOT / 'mattes').glob('*'))))
```

```
AIM images: 500
AIM mattes: 500
AIT images: 114
AIT mattes: 132
```

[56]:
```
split_and_copy(AIM_ROOT / 'original', AIM_ROOT / 'mask', split_ratios,␣
 ↪DATA_ROOT)
```

```

  Collected 500 valid pairs |   Renamed: 0
Found 500 pairs in /content/drive/MyDrive/AIM-500/original
1 copyingo_74234bc8.jpg
2 copyingo_dbef692f.jpg
3 copyingo_09b8ed47.jpg
4 copyingo_5a15b5f2.jpg
5 copyingo_77d7a529.jpg
6 copyingo_c0ca489f.jpg
7 copyingo_26b16002.jpg
8 copyingo_03d296a3.jpg
9 copyingo_ebc2e45c.jpg
10 copyingo_c060bd1c.jpg
11 copyingo_1c321c56.jpg
12 copyingo_71ed1e79.jpg
13 copyingo_669ed3b5.jpg
14 copyingo_093764ca.jpg
15 copyingo_660c8ca2.jpg
16 copyingo_2f033b3c.jpg
17 copyingo_cd5c068f.jpg
18 copyingo_5517109c.jpg
19 copyingo_e584f46f.jpg
20 copyingo_0530e2e7.jpg
21 copyingo_62f63948.jpg
22 copyingo_fae80c63.jpg
23 copyingo_418d6d35.jpg
24 copyingo_bb67ca50.jpg
25 copyingo_428276f8.jpg
26 copyingo_f07cde8a.jpg
```

```
27 copyingo_b085e7dc.jpg
28 copyingo_0a0ae43d.jpg
29 copyingo_bab88684.jpg
30 copyingo_3794e715.jpg
31 copyingo_7e3075f5.jpg
32 copyingo_9dbaf589.jpg
33 copyingo_f1ee35e0.jpg
34 copyingo_be63d1bf.jpg
35 copyingo_260f47c8.jpg
36 copyingo_e3fa38f7.jpg
37 copyingo_542332e9.jpg
38 copyingo_3b543a82.jpg
39 copyingo_e4f47cba.jpg
40 copyingo_6440e51e.jpg
41 copyingo_8b80a750.jpg
42 copyingo_50ebb2a6.jpg
43 copyingo_b286128e.jpg
44 copyingo_57684f7c.jpg
45 copyingo_69419c83.jpg
46 copyingo_7f028221.jpg
47 copyingo_d186fbeb.jpg
48 copyingo_84d5e4e7.jpg
49 copyingo_bf22c3be.jpg
50 copyingo_79c32126.jpg
51 copyingo_acc2727d.jpg
52 copyingo_a9d87400.jpg
53 copyingo_f6b02fc1.jpg
54 copyingo_4259400f.jpg
55 copyingo_16511265.jpg
56 copyingo_0bcaa573.jpg
57 copyingo_801135ce.jpg
58 copyingo_6be50c56.jpg
59 copyingo_876ab6b6.jpg
60 copyingo_e82f8b02.jpg
61 copyingo_f8ba7ef6.jpg
62 copyingo_1b224771.jpg
63 copyingo_9992c618.jpg
64 copyingo_e62df10b.jpg
65 copyingo_5851551d.jpg
66 copyingo_2c5d1d42.jpg
67 copyingo_e05960b7.jpg
68 copyingo_3579fa4c.jpg
69 copyingo_b208fdc3.jpg
70 copyingo_36848b17.jpg
71 copyingo_6b15071e.jpg
72 copyingo_73c97d7e.jpg
73 copyingo_62ecad5c.jpg
74 copyingo_54308893.jpg
```

```
75 copyingo_819ee421.jpg
76 copyingo_0d55695f.jpg
77 copyingo_c9160b3b.jpg
78 copyingo_247d445d.jpg
79 copyingo_1bedbb91.jpg
80 copyingo_e43c458b.jpg
81 copyingo_6be972db.jpg
82 copyingo_e4a72c06.jpg
83 copyingo_d983e00f.jpg
84 copyingo_b8665a4d.jpg
85 copyingo_a8b755f5.jpg
86 copyingo_ba76c547.jpg
87 copyingo_cf6cd65c.jpg
88 copyingo_69a1b2bd.jpg
89 copyingo_bde965af.jpg
90 copyingo_b4157744.jpg
91 copyingo_46886800.jpg
92 copyingo_b596f919.jpg
93 copyingo_34cddc11.jpg
94 copyingo_36fb7b85.jpg
95 copyingo_bd571707.jpg
96 copyingo_57993d37.jpg
97 copyingo_4729fa87.jpg
98 copyingo_c2cf00c3.jpg
99 copyingo_32846c6a.jpg
100 copyingo_0810cb44.jpg
101 copyingo_5a248199.jpg
102 copyingo_36d9b42a.jpg
103 copyingo_f515e5cf.jpg
104 copyingo_4cd1839f.jpg
105 copyingo_0a5e5a64.jpg
106 copyingo_bbf831ba.jpg
107 copyingo_f67dd514.jpg
108 copyingo_790de510.jpg
109 copyingo_74905c15.jpg
110 copyingo_740b4c5d.jpg
111 copyingo_404618b5.jpg
112 copyingo_4dbaa0fb.jpg
113 copyingo_b0c6c2f9.jpg
114 copyingo_22627075.jpg
115 copyingo_122b2e34.jpg
116 copyingo_76e29963.jpg
117 copyingo_2749e288.jpg
118 copyingo_57e4d780.jpg
119 copyingo_71dcc519.jpg
120 copyingo_009542a4.jpg
121 copyingo_a2b8a909.jpg
122 copyingo_2427e4a6.jpg
```

```
123 copyingo_79716548.jpg
124 copyingo_3c0b2617.jpg
125 copyingo_ec6becb3.jpg
126 copyingo_7c642e86.jpg
127 copyingo_494867da.jpg
128 copyingo_5d53cc84.jpg
129 copyingo_82d2a9b9.jpg
130 copyingo_42840dbf.jpg
131 copyingo_1f836c45.jpg
132 copyingo_6449cd22.jpg
133 copyingo_bc501d4e.jpg
134 copyingo_a5ba6f84.jpg
135 copyingo_d1c03e1c.jpg
136 copyingo_e0ab2760.jpg
137 copyingo_86527a2b.jpg
138 copyingo_4f134e05.jpg
139 copyingo_3a9a5e7c.jpg
140 copyingo_dde31d9f.jpg
141 copyingo_029a8efe.jpg
142 copyingo_fa41eb61.jpg
143 copyingo_df5b9e15.jpg
144 copyingo_a1513d78.jpg
145 copyingo_7f2daed3.jpg
146 copyingo_2a4715ae.jpg
147 copyingo_2edac560.jpg
148 copyingo_780f404a.jpg
149 copyingo_3890ea10.jpg
150 copyingo_66d830ec.jpg
151 copyingo_90c028ce.jpg
152 copyingo_b3b7ad01.jpg
153 copyingo_004cddc9.jpg
154 copyingo_a5038aad.jpg
155 copyingo_87e77b32.jpg
156 copyingo_197d7af1.jpg
157 copyingo_9405fe2d.jpg
158 copyingo_d45e64f1.jpg
159 copyingo_cfffcfb5.jpg
160 copyingo_81df2025.jpg
161 copyingo_7c8926da.jpg
162 copyingo_3312a26e.jpg
163 copyingo_6d96d978.jpg
164 copyingo_5031923f.jpg
165 copyingo_12b8b946.jpg
166 copyingo_84860d09.jpg
167 copyingo_527e5a54.jpg
168 copyingo_71f98381.jpg
169 copyingo_6d345648.jpg
170 copyingo_d2a1bff3.jpg
```

```
171 copyingo_c4fce710.jpg
172 copyingo_f20f853b.jpg
173 copyingo_1467861e.jpg
174 copyingo_6cd9e070.jpg
175 copyingo_27a6c2d7.jpg
176 copyingo_dd0868e0.jpg
177 copyingo_87ed5d9e.jpg
178 copyingo_2be0f6c9.jpg
179 copyingo_8215b759.jpg
180 copyingo_8ca72949.jpg
181 copyingo_94afaf0a.jpg
182 copyingo_2941a07a.jpg
183 copyingo_2d4043cb.jpg
184 copyingo_1a9abc07.jpg
185 copyingo_b62c06fe.jpg
186 copyingo_b061b5d7.jpg
187 copyingo_db5e685f.jpg
188 copyingo_56af7329.jpg
189 copyingo_5e83cbbe.jpg
190 copyingo_e30ce6cb.jpg
191 copyingo_f67a4a56.jpg
192 copyingo_1546f28b.jpg
193 copyingo_376c6d94.jpg
194 copyingo_9a3b9f8a.jpg
195 copyingo_a7df1f8f.jpg
196 copyingo_f937f968.jpg
197 copyingo_6e4a7373.jpg
198 copyingo_9f06988d.jpg
199 copyingo_b6b1fbea.jpg
200 copyingo_0b376f13.jpg
201 copyingo_f12a8afc.jpg
202 copyingo_3e213bb1.jpg
203 copyingo_34d216ed.jpg
204 copyingo_152611ad.jpg
205 copyingo_5d14b7e3.jpg
206 copyingo_e829249e.jpg
207 copyingo_663090ad.jpg
208 copyingo_52a2115b.jpg
209 copyingo_588650fe.jpg
210 copyingo_da41b554.jpg
211 copyingo_b40d228a.jpg
212 copyingo_443e89d1.jpg
213 copyingo_ff3d4315.jpg
214 copyingo_610852d2.jpg
215 copyingo_6fe67986.jpg
216 copyingo_9fb57e99.jpg
217 copyingo_16740854.jpg
218 copyingo_730a673b.jpg
```

```
219 copyingo_fe2c82a9.jpg
220 copyingo_3e33c72d.jpg
221 copyingo_aacf1d5c.jpg
222 copyingo_90862fb0.jpg
223 copyingo_57fd5490.jpg
224 copyingo_21c101d5.jpg
225 copyingo_f71b4ba5.jpg
226 copyingo_22dc01b9.jpg
227 copyingo_adfe7dcc.jpg
228 copyingo_099cc48f.jpg
229 copyingo_0723d28f.jpg
230 copyingo_f0afb039.jpg
231 copyingo_df40d3d0.jpg
232 copyingo_3b6f37e9.jpg
233 copyingo_d5c111c4.jpg
234 copyingo_1f05f13c.jpg
235 copyingo_599b2611.jpg
236 copyingo_b7c72c0a.jpg
237 copyingo_aedf5701.jpg
238 copyingo_684f0dfb.jpg
239 copyingo_8f254dc3.jpg
240 copyingo_e7597648.jpg
241 copyingo_1b6adf19.jpg
242 copyingo_97a204f5.jpg
243 copyingo_f7c5c922.jpg
244 copyingo_28ad039c.jpg
245 copyingo_ddae6d29.jpg
246 copyingo_7000c826.jpg
247 copyingo_053490ad.jpg
248 copyingo_4faab6b7.jpg
249 copyingo_26268b4b.jpg
250 copyingo_72a1f9ed.jpg
251 copyingo_33a4da38.jpg
252 copyingo_399eae2a.jpg
253 copyingo_12c828bc.jpg
254 copyingo_c46e3ddb.jpg
255 copyingo_9166152f.jpg
256 copyingo_bbcef72a.jpg
257 copyingo_8e2eb72f.jpg
258 copyingo_f22363ac.jpg
259 copyingo_6c3d1433.jpg
260 copyingo_b6b2771f.jpg
261 copyingo_e5e50a19.jpg
262 copyingo_4622cbb8.jpg
263 copyingo_acbecac7.jpg
264 copyingo_8a66ede1.jpg
265 copyingo_7f008485.jpg
266 copyingo_2972dbc2.jpg
```

```
267 copyingo_e92f575c.jpg
268 copyingo_fbae5321.jpg
269 copyingo_386e1cc3.jpg
270 copyingo_65e532ef.jpg
271 copyingo_d41b424a.jpg
272 copyingo_d29dea14.jpg
273 copyingo_aeb6180f.jpg
274 copyingo_57102043.jpg
275 copyingo_58b0c79e.jpg
276 copyingo_a974d3a3.jpg
277 copyingo_8431ecb9.jpg
278 copyingo_f95689c1.jpg
279 copyingo_706c33eb.jpg
280 copyingo_f4c2ba11.jpg
281 copyingo_76b23f51.jpg
282 copyingo_ad03d026.jpg
283 copyingo_e131f510.jpg
284 copyingo_35d8df37.jpg
285 copyingo_af47fb58.jpg
286 copyingo_76c7dc77.jpg
287 copyingo_70f45df9.jpg
288 copyingo_44dd34a4.jpg
289 copyingo_2c03ea4e.jpg
290 copyingo_1ae3ae29.jpg
291 copyingo_507a9a58.jpg
292 copyingo_ecd1265b.jpg
293 copyingo_61e47e14.jpg
294 copyingo_9d777ce8.jpg
295 copyingo_479bb899.jpg
296 copyingo_8c19e1a8.jpg
297 copyingo_cbe01a88.jpg
298 copyingo_162b27f9.jpg
299 copyingo_5c1f8c64.jpg
300 copyingo_0c33063a.jpg
301 copyingo_1c640020.jpg
302 copyingo_23bbcc9d.jpg
303 copyingo_125071d5.jpg
304 copyingo_915eb7f8.jpg
305 copyingo_90a7597b.jpg
306 copyingo_ca9317f9.jpg
307 copyingo_c8c91101.jpg
308 copyingo_62645d10.jpg
309 copyingo_8c31f783.jpg
310 copyingo_0bf712f6.jpg
311 copyingo_52aadf1d.jpg
312 copyingo_bfe0af9f.jpg
313 copyingo_be34e76b.jpg
314 copyingo_fe6f4047.jpg
```

```
315 copyingo_0594a762.jpg
316 copyingo_f6cd2fca.jpg
317 copyingo_d103af2f.jpg
318 copyingo_7b1db264.jpg
319 copyingo_a9cac469.jpg
320 copyingo_3c65e352.jpg
321 copyingo_1462ee19.jpg
322 copyingo_fabaf6e3.jpg
323 copyingo_263a94b8.jpg
324 copyingo_c6ef7a25.jpg
325 copyingo_4852f7d4.jpg
326 copyingo_a0c1d0d2.jpg
327 copyingo_8ede15d6.jpg
328 copyingo_98c4c527.jpg
329 copyingo_f82f042a.jpg
330 copyingo_e8ce31a6.jpg
331 copyingo_cc164147.jpg
332 copyingo_642f96e7.jpg
333 copyingo_4abc8a81.jpg
334 copyingo_1ab9353d.jpg
335 copyingo_44068300.jpg
336 copyingo_46a94998.jpg
337 copyingo_6aad0ebc.jpg
338 copyingo_1eb22a49.jpg
339 copyingo_214a5da4.jpg
340 copyingo_e92b90fc.jpg
341 copyingo_af1e10f2.jpg
342 copyingo_6372e660.jpg
343 copyingo_ccf42a69.jpg
344 copyingo_9d3a710b.jpg
345 copyingo_c24515b4.jpg
346 copyingo_4f7b0699.jpg
347 copyingo_798a02d7.jpg
348 copyingo_7e8fdc47.jpg
349 copyingo_c665e90e.jpg
350 copyingo_0b7228ec.jpg
1 copyingo_8a402a51.jpg
2 copyingo_e28df866.jpg
3 copyingo_11fef2b7.jpg
4 copyingo_91141ea8.jpg
5 copyingo_bd1e0dfc.jpg
6 copyingo_c15a96a6.jpg
7 copyingo_be721ca0.jpg
8 copyingo_de0bc50c.jpg
9 copyingo_1b0bfdee.jpg
10 copyingo_045bfd05.jpg
11 copyingo_42cc3714.jpg
12 copyingo_46818bea.jpg
```

```
13 copyingo_72d7b2ec.jpg
14 copyingo_561c2a05.jpg
15 copyingo_0df5178f.jpg
16 copyingo_73696a44.jpg
17 copyingo_2b0e2eed.jpg
18 copyingo_aa42b6b9.jpg
19 copyingo_268a126b.jpg
20 copyingo_202fb858.jpg
21 copyingo_e944bfaa.jpg
22 copyingo_8151559f.jpg
23 copyingo_2309157d.jpg
24 copyingo_e78005d6.jpg
25 copyingo_741da4bb.jpg
26 copyingo_8e17d667.jpg
27 copyingo_43fec7a1.jpg
28 copyingo_c5fb15fd.jpg
29 copyingo_16bf19b5.jpg
30 copyingo_7e259686.jpg
31 copyingo_5219d2ba.jpg
32 copyingo_6be8474d.jpg
33 copyingo_67f36f9b.jpg
34 copyingo_9c2968c9.jpg
35 copyingo_51f5ae17.jpg
36 copyingo_51873814.jpg
37 copyingo_14cfc225.jpg
38 copyingo_659c5db8.jpg
39 copyingo_3715959a.jpg
40 copyingo_b8de8b4b.jpg
41 copyingo_fb66f6f9.jpg
42 copyingo_f96afe15.jpg
43 copyingo_62ae16d7.jpg
44 copyingo_ce5d20a0.jpg
45 copyingo_3c7ed100.jpg
46 copyingo_3813eacb.jpg
47 copyingo_fad87b9e.jpg
48 copyingo_d74e3ad1.jpg
49 copyingo_380bb319.jpg
50 copyingo_d8b96fe7.jpg
51 copyingo_1edbc402.jpg
52 copyingo_8e8380f0.jpg
53 copyingo_aa56304c.jpg
54 copyingo_a73b4983.jpg
55 copyingo_ca026994.jpg
56 copyingo_8561c976.jpg
57 copyingo_7ffe1249.jpg
58 copyingo_985513ba.jpg
59 copyingo_751bcc44.jpg
60 copyingo_53100c73.jpg
```

```
61 copyingo_80193d44.jpg
62 copyingo_59db95e2.jpg
63 copyingo_b93c74e0.jpg
64 copyingo_4c33b91e.jpg
65 copyingo_f3cc6fa1.jpg
66 copyingo_26afdcad.jpg
67 copyingo_e225283a.jpg
68 copyingo_60104376.jpg
69 copyingo_c074bd23.jpg
70 copyingo_a3d4c2a6.jpg
71 copyingo_32e8ba1e.jpg
72 copyingo_10c239b7.jpg
73 copyingo_33d1515f.jpg
74 copyingo_a77d35f1.jpg
75 copyingo_7c88f64e.jpg
1 copyingo_ba1efe3b.jpg
2 copyingo_b2dfc20d.jpg
3 copyingo_2b9227eb.jpg
4 copyingo_776a8d45.jpg
5 copyingo_0fc8d771.jpg
6 copyingo_6b25213e.jpg
7 copyingo_8861e554.jpg
8 copyingo_cc63fd4a.jpg
9 copyingo_5313b681.jpg
10 copyingo_c01ea2f5.jpg
11 copyingo_a3798f05.jpg
12 copyingo_82a97798.jpg
13 copyingo_c56a90a4.jpg
14 copyingo_f0eaeebf.jpg
15 copyingo_1ea2b894.jpg
16 copyingo_5dad52d4.jpg
17 copyingo_7ccbf2cd.jpg
18 copyingo_0a09b978.jpg
19 copyingo_d2d8aa47.jpg
20 copyingo_8f8eb959.jpg
21 copyingo_14a9d723.jpg
22 copyingo_5e7d1db9.jpg
23 copyingo_b71d875b.jpg
24 copyingo_f3d0f881.jpg
25 copyingo_790c2756.jpg
26 copyingo_bf7d47a4.jpg
27 copyingo_b508d293.jpg
28 copyingo_db8321fa.jpg
29 copyingo_43ebe1e3.jpg
30 copyingo_b6f9703c.jpg
31 copyingo_ff3c576e.jpg
32 copyingo_aaa92d0a.jpg
33 copyingo_385e7ee8.jpg
```

```
34 copyingo_67b293e1.jpg
35 copyingo_7a796168.jpg
36 copyingo_fee0e426.jpg
37 copyingo_b1f17a6a.jpg
38 copyingo_246bd3c3.jpg
39 copyingo_69cc3268.jpg
40 copyingo_1b4c1dfc.jpg
41 copyingo_bea06fbe.jpg
42 copyingo_402b3954.jpg
43 copyingo_32b692c3.jpg
44 copyingo_b4074319.jpg
45 copyingo_912d43be.jpg
46 copyingo_508fa8da.jpg
47 copyingo_dc288b1a.jpg
48 copyingo_7fe72892.jpg
49 copyingo_da534991.jpg
50 copyingo_fda66b96.jpg
51 copyingo_c30a66c6.jpg
52 copyingo_cd0a70e6.jpg
53 copyingo_84eeb5ed.jpg
54 copyingo_9666a981.jpg
55 copyingo_f2347529.jpg
56 copyingo_8d099c35.jpg
57 copyingo_6e470e4a.jpg
58 copyingo_9109fee5.jpg
59 copyingo_a31a0472.jpg
60 copyingo_701f4571.jpg
61 copyingo_0c8f03f0.jpg
62 copyingo_9539f16c.jpg
63 copyingo_85ac3d9a.jpg
64 copyingo_d776b3ca.jpg
65 copyingo_dfcf2b98.jpg
66 copyingo_766d8742.jpg
67 copyingo_eaeab68d.jpg
68 copyingo_e2a6a947.jpg
69 copyingo_a3132475.jpg
70 copyingo_ec18734e.jpg
71 copyingo_c795d7e3.jpg
72 copyingo_b39cb6e4.jpg
73 copyingo_433595f0.jpg
74 copyingo_a7425b42.jpg
75 copyingo_868c53f0.jpg
```

[57]:

```
split_and_copy(AIT_ROOT / 'images', AIT_ROOT / 'mattes', split_ratios,␣
 ↪DATA_ROOT)
```

```
Renamed: Attaphongse Taparugssanage.png
Renamed: Rajeshwari B.png
```

```
Renamed: Mrs. Yashaswi H. V.png
Renamed: Dr. Avirut Puttiwongrak.png
Renamed: Prof.Avishek Datta.png
Renamed: Dr. Arlene Lu-Gonzales.png
Renamed: Andrew20Macintosh_edited.png
Renamed: Prof.Ajit P.Annachhatre.png
Renamed: Dr. Amararatne Yakupitiyage.png
Renamed: Dr. Ayushman Bhatt.png
Renamed: Mr. Benjamin Quinlan.png
Renamed: Prof. Bonaventura H. W. Hadikusumo.png
Renamed: Dr. Branesh Madhavan.png
Renamed: Dr. Chaklam Silpasuwanchai.png
Renamed: Dr.Chantri-Profile-Photo-For-Website-e1704251676272.png
Renamed: Mr. Chaarnkurt Yaoyuenyong.png
Renamed: Prof. Chettiyappan Visvanathan.png
Renamed: Prof. Chongrak Polprasert.png
Renamed: Dr. Chutiporn Anutariya.png
Renamed: Prof. Deepak Sharma.png
Renamed: Prof. Dieter Trau.png
Renamed: Dr. Djoen San Santoso.png
Renamed: Dr. Ekbordin Winijkul.png
Renamed: Duke-Malan.png
Renamed: Dr. Farhad Zulfiqar.png
Renamed: Prof. Gajendra Sigh.png
Renamed: Dr. Gerard Tocquer.png
Renamed: Dr. Ha Thanh Dong.png
Renamed: Dr. Hayat Ullah.png
Renamed: Prof. Huynh Trung Luong.png
Renamed: Dr. Indrajit Pal.png
Renamed: Prof. Jai Govind Singh.png
Renamed: Ms. Jarunee Wonlimpiyarat.png
Renamed: Prof. Jayant K. Routray.png
Renamed: Dr. Jitti Mungkalasiri.png
Renamed: Prof. Joyashree Roy.png
Renamed: Dr. Joyee S. Chatterjee.png
Renamed: Dr. Julaikha Bente Hossain.png
Renamed: Prof. Karl E. Weber.png
Renamed: Dr. Khin Myat Kyaw.png
Renamed: Mr. Kevin Pereira.png
Renamed: Dr. Kittipong Kkkachai.png
Renamed: Dr. Krishna R. Salin.png
Renamed: Prof. Kunnawee Kanitpong.png
Renamed: Dr. Kuo-Chieh Chao.png
Renamed: Prof. Kyoko Kusakabe.png
Renamed: Dr. Lakeesha K. Ransom.png
Renamed: Dr. Lonn Pichdara.png
Renamed: Prof. Manukid Parnichkun.png
Renamed: Dr. Mohana Sundaram Shanmugam.png
```

```
Renamed: Prof. Mokbul Morshed Ahmad.png
Renamed: Dr. Muhammad Junai.png
Renamed: Dr. Muhammad Yaseen.png
Renamed: Prof. Mukand Singh Babel.png
Renamed: Dr. Natthachet Tangdamrongsub.png
Renamed: Prof. Nguyen Thi Kim Onah.png
Renamed: Mr. Neil Doncaster.png
Renamed: Prof. Nophea Sasaki.png
Renamed: Dr. Nuwong chollacoop.png
Renamed: Dr. Oleg V. Shipin.png
Renamed: Ms. Paige Dobkin.png
Renamed: Dr. Panon Latcharote.png
Renamed: Prof. Paula Banerjee.png
Renamed: Prof. Pennung Warnitchai.png
Renamed: Prof. Peter Edwards.png
Renamed: Dr. Philipee Doneys.png
Renamed: Dr. Pished Bunnun.png
Renamed: Dr. Pisut Koomsap.png
Renamed: Prof. Nitin Kumar Tripathi.png
Renamed: Prof. P. Abdul Salam.png
Renamed: Prof. Sangam Shrestha.png
Renamed: Prof. Yosre Badir.png
Renamed: Dr. Punchet Thammarak.png
Renamed: Dr. Punchet Thammarak - Copy.png
Renamed: Dr. Raffaele Ricco.png
Renamed: Prof. Rajendra P. Shrestha.png
Renamed: Dr. Raktipong Sahamitmongkol - Copy.png
Renamed: Dr. Raktipong Sahamitmongkol.png
Renamed: Dr. Ram C. Bhujel.png
Renamed: Prof. Ram Shrestha.png
Renamed: Dr. Roger Levermore.png
Renamed: Dr. Roger Levermore - Copy.png
Renamed: Dr. Sakul Pochanart.png
Renamed: Dr. Salvatore G.P. Virdis.png
Renamed: Dr. Sarawut Ninsawat.png
Renamed: Dr. Sasitorn Srisawadi.png
Renamed: Dr. Sharaniya Vijitharan.png
Renamed: Prof. Shobhakar Dhakal.png
Renamed: Dr. Sitthisuntorn.png
Renamed: Ms. Stacey Huang.png
Renamed: Prof. Sivanappan Kumar.png
Renamed: Dr. Supasith Chonglerttham.png
Renamed: Dr. Sushil Kumar Himanshu.png
Renamed: Dr. Sutat Weesakul.png
Renamed: Dr. Suwussa Bamrungsap.png
Renamed: Dr. Takuji W. Tsusaka.png
Renamed: Dr. Tranujjal Bora.png
Renamed: Dr. Teerapat Sanguankotchakorn.png
```

```
  Renamed: Prof. Thammarat Koottatep.png
  Renamed: Prof. thanakorn Pheeraphan.png
  Renamed: Dr. Thi Phuoc Lai Nguyen.png
  Renamed: Dr. Tobias Endress.png
  Renamed: Dr. Uracha Ruktanonchai.png
  Renamed: Dr. Vatcharapol Sukhotu.png
  Renamed: Dr. Vatcharaporn Esichaikul.png
  Renamed: Prof. Vilas Nitivattananon.png
  Renamed: Mr. Ville Kulmala.png
  Renamed: Prof. Vinod Jindal.png
  Renamed: Dr. Wai Prathumpai.png
  Renamed: Prof. Weerakorn Ongsakul.png
  Renamed: Dr. Wenchao Xue.png
  Renamed: Dr. Willi Zimmermann.png
  Renamed: Dr. Wonnop Vissesanguan.png
  No matching matte found for: Prof. Mongkol Ekpanyapong.jpg

  Collected 113 valid pairs |  Renamed: 113
Found 113 pairs in /content/drive/MyDrive/AIT_Faculty/images
1 copyingProf. Vilas Nitivattananon.jpg
2 copyingProf. Yosre Badir.jpg
3 copyingDr. Sitthisuntorn.jpg
4 copyingDr. Sutat Weesakul.jpg
5 copyingDr. Ram C. Bhujel.jpg
6 copyingAndrew20Macintosh_edited.jpg
7 copyingDr. Hayat Ullah.jpg
8 copyingDr. Chutiporn Anutariya.png
9 copyingProf. Deepak Sharma.jpg
10 copyingDr. Sarawut Ninsawat.jpg
11 copyingDr. Philipee Doneys.jpg
12 copyingDr. Gerard Tocquer.jpg
13 copyingDr. Raffaele Ricco.jpg
14 copyingRajeshwari B.jpg
15 copyingDr. Thi Phuoc Lai Nguyen.jpg
16 copyingDr. Raktipong Sahamitmongkol.jpg
17 copyingProf. Gajendra Sigh.jpeg
18 copyingDr. Takuji W. Tsusaka.jpeg
19 copyingProf. Bonaventura H. W. Hadikusumo.jpg
20 copyingProf. Mokbul Morshed Ahmad.jpg
21 copyingProf. Vinod Jindal.jpeg
22 copyingProf. Ram Shrestha.jpeg
23 copyingMr. Chaarnkurt Yaoyuenyong.jpg
24 copyingProf. Kunnawee Kanitpong.jpg
25 copyingDr. Branesh Madhavan.jpeg
26 copyingProf.Avishek Datta.jpg
27 copyingProf. Sangam Shrestha.jpg
28 copyingProf.Ajit P.Annachhatre.jpeg
29 copyingProf. Jayant K. Routray.jpeg
```

```
30 copyingDr. Muhammad Junai.jpg
31 copyingProf. Rajendra P. Shrestha.jpeg
32 copyingProf. Kyoko Kusakabe.jpg
33 copyingDuke-Malan.jpg
34 copyingDr. Punchet Thammarak.jpeg
35 copyingDr. Willi Zimmermann.jpg
36 copyingDr. Krishna R. Salin.jpeg
37 copyingDr. Joyee S. Chatterjee.jpg
38 copyingDr. Chaklam Silpasuwanchai.jpg
39 copyingDr. Sasitorn Srisawadi.jpg
40 copyingMr. Benjamin Quinlan.jpg
41 copyingMs. Paige Dobkin.jpeg
42 copyingDr. Roger Levermore.jpg
43 copyingProf. Nophea Sasaki.jpg
44 copyingDr. Farhad Zulfiqar.jpg
45 copyingDr. Suwussa Bamrungsap.jpeg
46 copyingProf. Karl E. Weber.jpg
47 copyingProf. Chongrak Polprasert.jpeg
48 copyingDr. Pisut Koomsap.jpg
49 copyingDr. Mohana Sundaram Shanmugam.jpg
50 copyingDr. Sushil Kumar Himanshu.jpeg
51 copyingDr. Wenchao Xue.jpeg
52 copyingMr. Ville Kulmala.jpg
53 copyingDr. Wonnop Vissesanguan.jpg
54 copyingDr. Khin Myat Kyaw.jpg
55 copyingDr. Natthachet Tangdamrongsub.jpg
56 copyingAttaphongse Taparugssanage.jpg
57 copyingDr. Nuwong chollacoop.jpeg
58 copyingDr. Djoen San Santoso.jpg
59 copyingProf. Paula Banerjee.jpg
60 copyingProf. Weerakorn Ongsakul.jpeg
61 copyingDr. Teerapat Sanguankotchakorn.jpg
62 copyingDr. Lonn Pichdara.jpg
63 copyingDr. Avirut Puttiwongrak.jpg
64 copyingProf. Shobhakar Dhakal.jpeg
65 copyingDr. Tobias Endress.jpg
66 copyingMr. Neil Doncaster.jpg
67 copyingProf. Manukid Parnichkun.jpeg
68 copyingDr. Lakeesha K. Ransom.jpg
69 copyingDr. Indrajit Pal.jpg
70 copyingDr. Kuo-Chieh Chao.jpg
71 copyingDr. Wai Prathumpai.jpg
72 copyingMrs. Yashaswi H. V.jpg
73 copyingProf. Chettiyappan Visvanathan.jpg
74 copyingDr. Muhammad Yaseen.jpg
75 copyingDr. Julaikha Bente Hossain.jpg
76 copyingProf. Pennung Warnitchai.jpg
77 copyingDr. Sharaniya Vijitharan.jpg
```

```
78 copyingDr.Chantri-Profile-Photo-For-Website-e1704251676272.jpg
79 copyingDr. Roger Levermore - Copy.jpg
1 copyingDr. Ayushman Bhatt.jpg
2 copyingDr. Ha Thanh Dong.jpg
3 copyingDr. Kittipong Kkkachai.jpeg
4 copyingDr. Uracha Ruktanonchai.jpeg
5 copyingMr. Kevin Pereira.jpg
6 copyingProf. P. Abdul Salam.jpg
7 copyingProf. Peter Edwards.jpeg
8 copyingDr. Vatcharapol Sukhotu.jpeg
9 copyingDr. Ekbordin Winijkul.jpeg
10 copyingDr. Arlene Lu-Gonzales.jpg
11 copyingDr. Jitti Mungkalasiri.jpeg
12 copyingProf. Huynh Trung Luong.jpg
13 copyingProf. Dieter Trau.jpg
14 copyingDr. Oleg V. Shipin.jpg
15 copyingProf. Thammarat Koottatep.jpg
16 copyingDr. Raktipong Sahamitmongkol - Copy.jpg
1 copyingMs. Jarunee Wonlimpiyarat.jpg
2 copyingDr. Panon Latcharote.jpg
3 copyingProf. thanakorn Pheeraphan.jpg
4 copyingDr. Salvatore G.P. Virdis.jpg
5 copyingDr. Tranujjal Bora.jpg
6 copyingProf. Jai Govind Singh.jpeg
7 copyingProf. Joyashree Roy.jpg
8 copyingDr. Pished Bunnun.jpg
9 copyingProf. Nitin Kumar Tripathi.jpg
10 copyingProf. Mukand Singh Babel.jpg
11 copyingProf. Sivanappan Kumar.jpeg
12 copyingProf. Nguyen Thi Kim Onah.jpg
13 copyingDr. Supasith Chonglertham.jpg
14 copyingDr. Amararatne Yakupitiyage.jpeg
15 copyingDr. Punchet Thammarak - Copy.jpeg
16 copyingDr. Sakul Pochanart.jpg
17 copyingDr. Vatcharaporn Esichaikul.jpeg
18 copyingMs. Stacey Huang.jpg
```

[58]:
```python
from pathlib import Path

print("Train images:", len(list(Path(train_img_dir).glob('*'))))
print("Train mattes:", len(list(Path(train_matte_dir).glob('*'))))
print("Val images:", len(list(Path(val_img_dir).glob('*'))))
print("Val mattes:", len(list(Path(val_matte_dir).glob('*'))))
```

```
Train images: 429
Train mattes: 429
Val images: 91
Val mattes: 91
```

[59]:
```
import os, sys, importlib.util, types

# Path where you cloned the repo
repo_root = "/content/MODNet"
modnet_file = os.path.join(repo_root, "src", "models", "modnet.py")
backbones_file = os.path.join(repo_root, "src", "models", "backbones",␣
  ↪"__init__.py")

# Create stub packages so relative imports in modnet.py work
pkg_src = types.ModuleType("src"); pkg_src.__path__ = [os.path.join(repo_root,␣
  ↪"src")]
pkg_models = types.ModuleType("src.models"); pkg_models.__path__ = [os.path.
  ↪join(repo_root, "src", "models")]
sys.modules["src"] = pkg_src
sys.modules["src.models"] = pkg_models

# Load backbones first
spec_bk = importlib.util.spec_from_file_location("src.models.backbones",␣
  ↪backbones_file)
bk_mod = importlib.util.module_from_spec(spec_bk)
sys.modules["src.models.backbones"] = bk_mod
spec_bk.loader.exec_module(bk_mod)

# Load modnet
spec_md = importlib.util.spec_from_file_location("src.models.modnet",␣
  ↪modnet_file)
modnet_mod = importlib.util.module_from_spec(spec_md)
sys.modules["src.models.modnet"] = modnet_mod
spec_md.loader.exec_module(modnet_mod)

# Get class
MODNet = modnet_mod.MODNet
print("  MODNet class loaded:", MODNet)
```

```
 MODNet class loaded: <class 'src.models.modnet.MODNet'>
```

[60]:
```
# !mkdir -p /content/MODNet/pretrained
# !wget -q https://github.com/ZHKKKe/MODNet/releases/download/v1/
  ↪modnet_webcam_portrait_matting.ckpt \
#     -O /content/MODNet/pretrained/modnet_webcam_portrait_matting.ckpt
```

[61]:
```
from PIL import Image, UnidentifiedImageError
from pathlib import Path
import os

def remove_unidentified(folder_path):
    folder = Path(folder_path)
    removed = []

    for file in folder.glob("*"):
        try:
            Image.open(file).verify()
        except UnidentifiedImageError:
            print(f" Removing unreadable file: {file.name}")
            os.remove(file)
            removed.append(file.stem)  # store stem for cross-checking

    print(f" Removed {len(removed)} files from {folder.name}")
    return removed

# Define your paths
# img_dir = "/content/drive/MyDrive/modnet_data/train/images"
# matte_dir = "/content/drive/MyDrive/modnet_data/train/mattes"

# Remove unreadable files
bad_img_stems = remove_unidentified(train_img_dir)
bad_matte_stems = remove_unidentified(train_matte_dir)

# Optional: remove mismatched pairs
def remove_mismatched_pairs(img_dir, matte_dir, bad_stems):
    for stem in bad_stems:
        img_file = list(Path(img_dir).glob(stem + ".*"))
        matte_file = list(Path(matte_dir).glob(stem + ".*"))
        for f in img_file + matte_file:
            if f.exists():
                print(f" Removing mismatched file: {f.name}")
                os.remove(f)

# Remove any remaining mismatched pairs
remove_mismatched_pairs(train_img_dir, train_matte_dir, set(bad_img_stems +␣
 ↪bad_matte_stems))
```

```
 Removed 0 files from images
 Removed 0 files from mattes
```

[62]:

```
# ========================================================
#   Updated PortraitDataset (Fix Normalize Issue)
# ========================================================
class PortraitDataset(Dataset):
    def __init__(self, img_dir, matte_dir, img_transform=None,␣
 ↪matte_transform=None):
        self.img_dir = Path(img_dir)
        self.matte_dir = Path(matte_dir)
        self.img_transform = img_transform
        self.matte_transform = matte_transform

        img_stems = {p.stem for p in self.img_dir.glob('*')}
        matte_stems = {p.stem for p in self.matte_dir.glob('*')}
        self.common_stems = sorted(list(img_stems & matte_stems))
        print(f"  Found {len(self.common_stems)} paired samples in {self.
↪img_dir.parent.name}")

    def __len__(self):
        return len(self.common_stems)

    def __getitem__(self, idx):
          stem = self.common_stems[idx]

          img_matches = list(self.img_dir.glob(stem + '.*'))
          matte_matches = list(self.matte_dir.glob(stem + '.*'))

          if not img_matches or not matte_matches:
              print(f"  Missing file for stem: {stem}")
              return self.__getitem__((idx + 1) % len(self.common_stems))  #␣
↪fallback to next sample

          img_path = img_matches[0]
          matte_path = matte_matches[0]

          try:
              image = Image.open(img_path).convert("RGB")
              matte = Image.open(matte_path).convert("L")
          except UnidentifiedImageError:
              print(f"  Unreadable image or matte: {stem}")
              return self.__getitem__((idx + 1) % len(self.common_stems))

          if self.img_transform:
              image = self.img_transform(image)
          if self.matte_transform:
              matte = self.matte_transform(matte)

          return image, matte
```

[63]:

```
import torchvision.transforms as T

transform_img = T.Compose([
    T.Resize((512, 512)),
    T.ToTensor()
])

transform_matte = T.Compose([
    T.Resize((512, 512)),
    T.ToTensor()
])
```

[64]:
```
train_img_dir = Path('/content/drive/MyDrive/modnet_data/train/images')
train_matte_dir = Path('/content/drive/MyDrive/modnet_data/train/mattes')



print("Images folder exists:", train_img_dir.exists())
print("Mattes folder exists:", train_matte_dir.exists())
print("Number of image files:", len(list(train_img_dir.glob('*'))))
print("Number of matte files:", len(list(train_matte_dir.glob('*'))))
```

```
Images folder exists: True
Mattes folder exists: True
Number of image files: 429
Number of matte files: 429
```

[66]:
```
print("First 10 stems found in images:", sorted([p.stem for p in train_img_dir.
  ↪glob('*')])[:10])
print("First 10 stems found in mattes:", sorted([p.stem for p in␣
  ↪train_matte_dir.glob('*')])[:10])
```

```
First 10 stems found in images: ['Andrew20Macintosh_edited', 'Attaphongse
Taparugssanage', 'Dr. Avirut Puttiwongrak', 'Dr. Branesh Madhavan', 'Dr. Chaklam
Silpasuwanchai', 'Dr. Chutiporn Anutariya', 'Dr. Djoen San Santoso', 'Dr. Farhad
Zulfiqar', 'Dr. Gerard Tocquer', 'Dr. Hayat Ullah']
First 10 stems found in mattes: ['Andrew20Macintosh_edited', 'Attaphongse
Taparugssanage', 'Dr. Avirut Puttiwongrak', 'Dr. Branesh Madhavan', 'Dr. Chaklam
Silpasuwanchai', 'Dr. Chutiporn Anutariya', 'Dr. Djoen San Santoso', 'Dr. Farhad
Zulfiqar', 'Dr. Gerard Tocquer', 'Dr. Hayat Ullah']
```

[79]:
```
from torch.utils.data import DataLoader
from torchvision import transforms

# --- Image preprocessing ---
# Resize all samples to 512×512 (MODNet input size) and convert to tensors
# Normalization is optional here; MODNet uses range [-1, 1] internally.
transform_train = transforms.Compose([
    transforms.Resize((512, 512)),
    transforms.ToTensor(),
    transforms.Normalize(mean=[0.5, 0.5, 0.5], std=[0.5, 0.5, 0.5]),
])

transform_val = transforms.Compose([
    transforms.Resize((512, 512)),
    transforms.ToTensor(),
    transforms.Normalize(mean=[0.5, 0.5, 0.5], std=[0.5, 0.5, 0.5]),
])

# --- Create Dataset instances ---
train_ds = PortraitDataset(
    train_img_dir,
    train_matte_dir,
    img_transform=transform_train,
    # matte_transform=transform_matte_train  # optional
    matte_transform=transform_matte
)

val_ds = PortraitDataset(
    val_img_dir,
    val_matte_dir,
    img_transform=transform_val,
    # matte_transform=transform_matte_val  # optional
    matte_transform=transform_matte
)

# --- DataLoaders ---
train_loader = DataLoader(train_ds, batch_size=4, shuffle=True, num_workers=2,␣
 ↪pin_memory=True)
val_loader   = DataLoader(val_ds, batch_size=2, shuffle=False, num_workers=0,␣
 ↪pin_memory=True)


print(f" Train set: {len(train_ds)} samples, Val set: {len(val_ds)} samples")
```

```
Found 429 paired samples in train
Found 91 paired samples in val
Train set: 429 samples, Val set: 91 samples
```

[70]:

```
# ======================================================
#   Part 3 - Load Pretrained MODNet Model
# ======================================================
import torch
from pathlib import Path
import sys

# # --- Device setup ---
# device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')
# print(f" Using device: {device}")

# # --- Add MODNet source path ---
# # Make sure you cloned MODNet official repo or have the source in your␣
 ↪environment.
```

```
# MODNET_PATH = Path('/content/MODNet/src')  # adjust if needed
# if not MODNET_PATH.exists():
#     raise FileNotFoundError(" MODNet source path not found! Please clone it␣
 ↪into /content/MODNet")

# if str(MODNET_PATH) not in sys.path:
#     sys.path.append(str(MODNET_PATH))

# --- Import the model class ---
# from models.modnet import MODNet

# --- Initialize model ---
model = MODNet(backbone_pretrained=False).to(device)

# --- Load pretrained checkpoint ---
pretrained_model_name ='modnet_photographic_portrait_matting.ckpt'
PRETRAINED_CKPT = Path('/content/drive/MyDrive/modnet_data/pretrained/
 ↪'+pretrained_model_name)
print(f" Loading pretrained weights from: {PRETRAINED_CKPT}")

state = torch.load(PRETRAINED_CKPT, map_location=device)

# handle different checkpoint formats
if isinstance(state, dict) and 'state_dict' in state:
    state = state['state_dict']
state = {k.replace('module.', ''): v for k, v in state.items()}

missing, unexpected = model.load_state_dict(state, strict=False)
print(f" Model loaded. Missing: {len(missing)} | Unexpected:␣
 ↪{len(unexpected)}")

# --- Set to training mode ---
model.train()
```

```
 Loading pretrained weights from: /content/drive/MyDrive/modnet_data/pretrained
/modnet_photographic_portrait_matting.ckpt
 Model loaded. Missing: 0 | Unexpected: 0
```

[70]:

```
MODNet(
  (backbone): MobileNetV2Backbone(
    (model): MobileNetV2(
      (features): Sequential(
        (0): Sequential(
          (0): Conv2d(3, 32, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1),
bias=False)
          (1): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
```

```
          (2): ReLU6(inplace=True)
        )
        (1): InvertedResidual(
          (conv): Sequential(
            (0): Conv2d(32, 32, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), groups=32, bias=False)
            (1): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
            (2): ReLU6(inplace=True)
            (3): Conv2d(32, 16, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (4): BatchNorm2d(16, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          )
        )
        (2): InvertedResidual(
          (conv): Sequential(
            (0): Conv2d(16, 96, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (1): BatchNorm2d(96, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
            (2): ReLU6(inplace=True)
            (3): Conv2d(96, 96, kernel_size=(3, 3), stride=(2, 2), padding=(1,
1), groups=96, bias=False)
            (4): BatchNorm2d(96, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
            (5): ReLU6(inplace=True)
            (6): Conv2d(96, 24, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (7): BatchNorm2d(24, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          )
        )
        (3): InvertedResidual(
          (conv): Sequential(
            (0): Conv2d(24, 144, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (1): BatchNorm2d(144, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
            (2): ReLU6(inplace=True)
            (3): Conv2d(144, 144, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), groups=144, bias=False)
            (4): BatchNorm2d(144, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
            (5): ReLU6(inplace=True)
            (6): Conv2d(144, 24, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (7): BatchNorm2d(24, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          )
        )
        (4): InvertedResidual(
          (conv): Sequential(
            (0): Conv2d(24, 144, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (1): BatchNorm2d(144, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
            (2): ReLU6(inplace=True)
            (3): Conv2d(144, 144, kernel_size=(3, 3), stride=(2, 2), padding=(1,
1), groups=144, bias=False)
            (4): BatchNorm2d(144, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
            (5): ReLU6(inplace=True)
            (6): Conv2d(144, 32, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (7): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          )
        )
        (5): InvertedResidual(
          (conv): Sequential(
            (0): Conv2d(32, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (1): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
            (2): ReLU6(inplace=True)
            (3): Conv2d(192, 192, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), groups=192, bias=False)
            (4): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
            (5): ReLU6(inplace=True)
            (6): Conv2d(192, 32, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (7): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          )
        )
        (6): InvertedResidual(
          (conv): Sequential(
            (0): Conv2d(32, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (1): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
            (2): ReLU6(inplace=True)
            (3): Conv2d(192, 192, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), groups=192, bias=False)
            (4): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
            (5): ReLU6(inplace=True)
            (6): Conv2d(192, 32, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (7): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          )
        )
        (7): InvertedResidual(
          (conv): Sequential(
            (0): Conv2d(32, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (1): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
            (2): ReLU6(inplace=True)
            (3): Conv2d(192, 192, kernel_size=(3, 3), stride=(2, 2), padding=(1,
1), groups=192, bias=False)
            (4): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
            (5): ReLU6(inplace=True)
            (6): Conv2d(192, 64, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (7): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          )
        )
        (8): InvertedResidual(
          (conv): Sequential(
            (0): Conv2d(64, 384, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (1): BatchNorm2d(384, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
            (2): ReLU6(inplace=True)
            (3): Conv2d(384, 384, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), groups=384, bias=False)
            (4): BatchNorm2d(384, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
            (5): ReLU6(inplace=True)
            (6): Conv2d(384, 64, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (7): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          )
        )
        (9): InvertedResidual(
          (conv): Sequential(
            (0): Conv2d(64, 384, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (1): BatchNorm2d(384, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
            (2): ReLU6(inplace=True)
            (3): Conv2d(384, 384, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), groups=384, bias=False)
            (4): BatchNorm2d(384, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
            (5): ReLU6(inplace=True)
            (6): Conv2d(384, 64, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (7): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          )
        )
        (10): InvertedResidual(
          (conv): Sequential(
            (0): Conv2d(64, 384, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (1): BatchNorm2d(384, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
            (2): ReLU6(inplace=True)
            (3): Conv2d(384, 384, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), groups=384, bias=False)
            (4): BatchNorm2d(384, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
            (5): ReLU6(inplace=True)
            (6): Conv2d(384, 64, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (7): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          )
        )
        (11): InvertedResidual(
          (conv): Sequential(
            (0): Conv2d(64, 384, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (1): BatchNorm2d(384, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
            (2): ReLU6(inplace=True)
            (3): Conv2d(384, 384, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), groups=384, bias=False)
            (4): BatchNorm2d(384, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
            (5): ReLU6(inplace=True)
            (6): Conv2d(384, 96, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (7): BatchNorm2d(96, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          )
        )
        (12): InvertedResidual(
          (conv): Sequential(
            (0): Conv2d(96, 576, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (1): BatchNorm2d(576, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
            (2): ReLU6(inplace=True)
            (3): Conv2d(576, 576, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), groups=576, bias=False)
            (4): BatchNorm2d(576, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
            (5): ReLU6(inplace=True)
            (6): Conv2d(576, 96, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (7): BatchNorm2d(96, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
```

```
          )
        )
        (13): InvertedResidual(
          (conv): Sequential(
            (0): Conv2d(96, 576, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (1): BatchNorm2d(576, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
            (2): ReLU6(inplace=True)
            (3): Conv2d(576, 576, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), groups=576, bias=False)
            (4): BatchNorm2d(576, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
            (5): ReLU6(inplace=True)
            (6): Conv2d(576, 96, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (7): BatchNorm2d(96, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          )
        )
        (14): InvertedResidual(
          (conv): Sequential(
            (0): Conv2d(96, 576, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (1): BatchNorm2d(576, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
            (2): ReLU6(inplace=True)
            (3): Conv2d(576, 576, kernel_size=(3, 3), stride=(2, 2), padding=(1,
1), groups=576, bias=False)
            (4): BatchNorm2d(576, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
            (5): ReLU6(inplace=True)
            (6): Conv2d(576, 160, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (7): BatchNorm2d(160, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          )
        )
        (15): InvertedResidual(
          (conv): Sequential(
            (0): Conv2d(160, 960, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (1): BatchNorm2d(960, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
            (2): ReLU6(inplace=True)
            (3): Conv2d(960, 960, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), groups=960, bias=False)
            (4): BatchNorm2d(960, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
            (5): ReLU6(inplace=True)
            (6): Conv2d(960, 160, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (7): BatchNorm2d(160, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          )
        )
        (16): InvertedResidual(
          (conv): Sequential(
            (0): Conv2d(160, 960, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (1): BatchNorm2d(960, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
            (2): ReLU6(inplace=True)
            (3): Conv2d(960, 960, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), groups=960, bias=False)
            (4): BatchNorm2d(960, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
            (5): ReLU6(inplace=True)
            (6): Conv2d(960, 160, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (7): BatchNorm2d(160, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          )
        )
        (17): InvertedResidual(
          (conv): Sequential(
            (0): Conv2d(160, 960, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (1): BatchNorm2d(960, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
            (2): ReLU6(inplace=True)
            (3): Conv2d(960, 960, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), groups=960, bias=False)
            (4): BatchNorm2d(960, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
            (5): ReLU6(inplace=True)
            (6): Conv2d(960, 320, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (7): BatchNorm2d(320, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          )
        )
        (18): Sequential(
          (0): Conv2d(320, 1280, kernel_size=(1, 1), stride=(1, 1), bias=False)
          (1): BatchNorm2d(1280, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (2): ReLU6(inplace=True)
        )
      )
    )
  )
  (lr_branch): LRBranch(
    (backbone): MobileNetV2Backbone(
      (model): MobileNetV2(
        (features): Sequential(
          (0): Sequential(
            (0): Conv2d(3, 32, kernel_size=(3, 3), stride=(2, 2), padding=(1,
1), bias=False)
            (1): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
            (2): ReLU6(inplace=True)
          )
          (1): InvertedResidual(
            (conv): Sequential(
              (0): Conv2d(32, 32, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), groups=32, bias=False)
              (1): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
              (2): ReLU6(inplace=True)
              (3): Conv2d(32, 16, kernel_size=(1, 1), stride=(1, 1), bias=False)
              (4): BatchNorm2d(16, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (2): InvertedResidual(
            (conv): Sequential(
              (0): Conv2d(16, 96, kernel_size=(1, 1), stride=(1, 1), bias=False)
              (1): BatchNorm2d(96, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
              (2): ReLU6(inplace=True)
              (3): Conv2d(96, 96, kernel_size=(3, 3), stride=(2, 2), padding=(1,
1), groups=96, bias=False)
              (4): BatchNorm2d(96, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
              (5): ReLU6(inplace=True)
              (6): Conv2d(96, 24, kernel_size=(1, 1), stride=(1, 1), bias=False)
              (7): BatchNorm2d(24, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (3): InvertedResidual(
            (conv): Sequential(
              (0): Conv2d(24, 144, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(144, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
              (2): ReLU6(inplace=True)
              (3): Conv2d(144, 144, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), groups=144, bias=False)
              (4): BatchNorm2d(144, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
```

```
              (5): ReLU6(inplace=True)
              (6): Conv2d(144, 24, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (7): BatchNorm2d(24, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (4): InvertedResidual(
            (conv): Sequential(
              (0): Conv2d(24, 144, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(144, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
              (2): ReLU6(inplace=True)
              (3): Conv2d(144, 144, kernel_size=(3, 3), stride=(2, 2),
padding=(1, 1), groups=144, bias=False)
              (4): BatchNorm2d(144, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
              (5): ReLU6(inplace=True)
              (6): Conv2d(144, 32, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (7): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (5): InvertedResidual(
            (conv): Sequential(
              (0): Conv2d(32, 192, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
              (2): ReLU6(inplace=True)
              (3): Conv2d(192, 192, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), groups=192, bias=False)
              (4): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
              (5): ReLU6(inplace=True)
              (6): Conv2d(192, 32, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (7): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (6): InvertedResidual(
            (conv): Sequential(
              (0): Conv2d(32, 192, kernel_size=(1, 1), stride=(1, 1),
bias=False)
```

```
              (1): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
              (2): ReLU6(inplace=True)
              (3): Conv2d(192, 192, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), groups=192, bias=False)
              (4): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
              (5): ReLU6(inplace=True)
              (6): Conv2d(192, 32, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (7): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (7): InvertedResidual(
            (conv): Sequential(
              (0): Conv2d(32, 192, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
              (2): ReLU6(inplace=True)
              (3): Conv2d(192, 192, kernel_size=(3, 3), stride=(2, 2),
padding=(1, 1), groups=192, bias=False)
              (4): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
              (5): ReLU6(inplace=True)
              (6): Conv2d(192, 64, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (7): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (8): InvertedResidual(
            (conv): Sequential(
              (0): Conv2d(64, 384, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(384, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
              (2): ReLU6(inplace=True)
              (3): Conv2d(384, 384, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), groups=384, bias=False)
              (4): BatchNorm2d(384, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
              (5): ReLU6(inplace=True)
              (6): Conv2d(384, 64, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (7): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (9): InvertedResidual(
            (conv): Sequential(
              (0): Conv2d(64, 384, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(384, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
              (2): ReLU6(inplace=True)
              (3): Conv2d(384, 384, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), groups=384, bias=False)
              (4): BatchNorm2d(384, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
              (5): ReLU6(inplace=True)
              (6): Conv2d(384, 64, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (7): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (10): InvertedResidual(
            (conv): Sequential(
              (0): Conv2d(64, 384, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(384, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
              (2): ReLU6(inplace=True)
              (3): Conv2d(384, 384, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), groups=384, bias=False)
              (4): BatchNorm2d(384, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
              (5): ReLU6(inplace=True)
              (6): Conv2d(384, 64, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (7): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (11): InvertedResidual(
            (conv): Sequential(
              (0): Conv2d(64, 384, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(384, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
              (2): ReLU6(inplace=True)
              (3): Conv2d(384, 384, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), groups=384, bias=False)
              (4): BatchNorm2d(384, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
              (5): ReLU6(inplace=True)
              (6): Conv2d(384, 96, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (7): BatchNorm2d(96, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (12): InvertedResidual(
            (conv): Sequential(
              (0): Conv2d(96, 576, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(576, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
              (2): ReLU6(inplace=True)
              (3): Conv2d(576, 576, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), groups=576, bias=False)
              (4): BatchNorm2d(576, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
              (5): ReLU6(inplace=True)
              (6): Conv2d(576, 96, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (7): BatchNorm2d(96, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (13): InvertedResidual(
            (conv): Sequential(
              (0): Conv2d(96, 576, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(576, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
              (2): ReLU6(inplace=True)
              (3): Conv2d(576, 576, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), groups=576, bias=False)
              (4): BatchNorm2d(576, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
              (5): ReLU6(inplace=True)
              (6): Conv2d(576, 96, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (7): BatchNorm2d(96, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (14): InvertedResidual(
            (conv): Sequential(
              (0): Conv2d(96, 576, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(576, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
              (2): ReLU6(inplace=True)
              (3): Conv2d(576, 576, kernel_size=(3, 3), stride=(2, 2),
padding=(1, 1), groups=576, bias=False)
              (4): BatchNorm2d(576, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
              (5): ReLU6(inplace=True)
              (6): Conv2d(576, 160, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (7): BatchNorm2d(160, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (15): InvertedResidual(
            (conv): Sequential(
              (0): Conv2d(160, 960, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(960, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
              (2): ReLU6(inplace=True)
              (3): Conv2d(960, 960, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), groups=960, bias=False)
              (4): BatchNorm2d(960, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
              (5): ReLU6(inplace=True)
              (6): Conv2d(960, 160, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (7): BatchNorm2d(160, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (16): InvertedResidual(
            (conv): Sequential(
              (0): Conv2d(160, 960, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(960, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
              (2): ReLU6(inplace=True)
              (3): Conv2d(960, 960, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), groups=960, bias=False)
              (4): BatchNorm2d(960, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
              (5): ReLU6(inplace=True)
              (6): Conv2d(960, 160, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (7): BatchNorm2d(160, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (17): InvertedResidual(
            (conv): Sequential(
              (0): Conv2d(160, 960, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(960, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
              (2): ReLU6(inplace=True)
              (3): Conv2d(960, 960, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), groups=960, bias=False)
              (4): BatchNorm2d(960, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
              (5): ReLU6(inplace=True)
              (6): Conv2d(960, 320, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (7): BatchNorm2d(320, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (18): Sequential(
            (0): Conv2d(320, 1280, kernel_size=(1, 1), stride=(1, 1),
bias=False)
            (1): BatchNorm2d(1280, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
            (2): ReLU6(inplace=True)
          )
        )
      )
    )
    (se_block): SEBlock(
      (pool): AdaptiveAvgPool2d(output_size=1)
      (fc): Sequential(
        (0): Linear(in_features=1280, out_features=320, bias=False)
        (1): ReLU(inplace=True)
        (2): Linear(in_features=320, out_features=1280, bias=False)
        (3): Sigmoid()
      )
    )
    (conv_lr16x): Conv2dIBNormRelu(
      (layers): Sequential(
        (0): Conv2d(1280, 96, kernel_size=(5, 5), stride=(1, 1), padding=(2, 2))
        (1): IBNorm(
          (bnorm): BatchNorm2d(48, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (inorm): InstanceNorm2d(48, eps=1e-05, momentum=0.1, affine=False,
track_running_stats=False)
        )
        (2): ReLU(inplace=True)
      )
    )
    (conv_lr8x): Conv2dIBNormRelu(
      (layers): Sequential(
        (0): Conv2d(96, 32, kernel_size=(5, 5), stride=(1, 1), padding=(2, 2))
        (1): IBNorm(
          (bnorm): BatchNorm2d(16, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (inorm): InstanceNorm2d(16, eps=1e-05, momentum=0.1, affine=False,
track_running_stats=False)
        )
        (2): ReLU(inplace=True)
      )
    )
    (conv_lr): Conv2dIBNormRelu(
      (layers): Sequential(
        (0): Conv2d(32, 1, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1))
      )
    )
  )
  (hr_branch): HRBranch(
    (tohr_enc2x): Conv2dIBNormRelu(
      (layers): Sequential(
        (0): Conv2d(16, 32, kernel_size=(1, 1), stride=(1, 1))
        (1): IBNorm(
          (bnorm): BatchNorm2d(16, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (inorm): InstanceNorm2d(16, eps=1e-05, momentum=0.1, affine=False,
track_running_stats=False)
        )
        (2): ReLU(inplace=True)
      )
    )
    (conv_enc2x): Conv2dIBNormRelu(
      (layers): Sequential(
        (0): Conv2d(35, 32, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1))
        (1): IBNorm(
          (bnorm): BatchNorm2d(16, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (inorm): InstanceNorm2d(16, eps=1e-05, momentum=0.1, affine=False,
track_running_stats=False)
```

```
        )
        (2): ReLU(inplace=True)
      )
    )
    (tohr_enc4x): Conv2dIBNormRelu(
      (layers): Sequential(
        (0): Conv2d(24, 32, kernel_size=(1, 1), stride=(1, 1))
        (1): IBNorm(
          (bnorm): BatchNorm2d(16, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (inorm): InstanceNorm2d(16, eps=1e-05, momentum=0.1, affine=False,
track_running_stats=False)
        )
        (2): ReLU(inplace=True)
      )
    )
    (conv_enc4x): Conv2dIBNormRelu(
      (layers): Sequential(
        (0): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
        (1): IBNorm(
          (bnorm): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (inorm): InstanceNorm2d(32, eps=1e-05, momentum=0.1, affine=False,
track_running_stats=False)
        )
        (2): ReLU(inplace=True)
      )
    )
    (conv_hr4x): Sequential(
      (0): Conv2dIBNormRelu(
        (layers): Sequential(
          (0): Conv2d(99, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
          (1): IBNorm(
            (bnorm): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
            (inorm): InstanceNorm2d(32, eps=1e-05, momentum=0.1, affine=False,
track_running_stats=False)
          )
          (2): ReLU(inplace=True)
        )
      )
      (1): Conv2dIBNormRelu(
        (layers): Sequential(
          (0): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
          (1): IBNorm(
            (bnorm): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
```

```
            (inorm): InstanceNorm2d(32, eps=1e-05, momentum=0.1, affine=False,
track_running_stats=False)
          )
          (2): ReLU(inplace=True)
        )
      )
      (2): Conv2dIBNormRelu(
        (layers): Sequential(
          (0): Conv2d(64, 32, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
          (1): IBNorm(
            (bnorm): BatchNorm2d(16, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
            (inorm): InstanceNorm2d(16, eps=1e-05, momentum=0.1, affine=False,
track_running_stats=False)
          )
          (2): ReLU(inplace=True)
        )
      )
    )
    (conv_hr2x): Sequential(
      (0): Conv2dIBNormRelu(
        (layers): Sequential(
          (0): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
          (1): IBNorm(
            (bnorm): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
            (inorm): InstanceNorm2d(32, eps=1e-05, momentum=0.1, affine=False,
track_running_stats=False)
          )
          (2): ReLU(inplace=True)
        )
      )
      (1): Conv2dIBNormRelu(
        (layers): Sequential(
          (0): Conv2d(64, 32, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
          (1): IBNorm(
            (bnorm): BatchNorm2d(16, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
            (inorm): InstanceNorm2d(16, eps=1e-05, momentum=0.1, affine=False,
track_running_stats=False)
          )
          (2): ReLU(inplace=True)
        )
      )
      (2): Conv2dIBNormRelu(
        (layers): Sequential(
          (0): Conv2d(32, 32, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
          (1): IBNorm(
            (bnorm): BatchNorm2d(16, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
            (inorm): InstanceNorm2d(16, eps=1e-05, momentum=0.1, affine=False,
track_running_stats=False)
          )
          (2): ReLU(inplace=True)
        )
      )
      (3): Conv2dIBNormRelu(
        (layers): Sequential(
          (0): Conv2d(32, 32, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
          (1): IBNorm(
            (bnorm): BatchNorm2d(16, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
            (inorm): InstanceNorm2d(16, eps=1e-05, momentum=0.1, affine=False,
track_running_stats=False)
          )
          (2): ReLU(inplace=True)
        )
      )
    )
    (conv_hr): Sequential(
      (0): Conv2dIBNormRelu(
        (layers): Sequential(
          (0): Conv2d(35, 32, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
          (1): IBNorm(
            (bnorm): BatchNorm2d(16, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
            (inorm): InstanceNorm2d(16, eps=1e-05, momentum=0.1, affine=False,
track_running_stats=False)
          )
          (2): ReLU(inplace=True)
        )
      )
      (1): Conv2dIBNormRelu(
        (layers): Sequential(
          (0): Conv2d(32, 1, kernel_size=(1, 1), stride=(1, 1))
        )
      )
    )
  )
  (f_branch): FusionBranch(
    (conv_lr4x): Conv2dIBNormRelu(
      (layers): Sequential(
        (0): Conv2d(32, 32, kernel_size=(5, 5), stride=(1, 1), padding=(2, 2))
        (1): IBNorm(
          (bnorm): BatchNorm2d(16, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (inorm): InstanceNorm2d(16, eps=1e-05, momentum=0.1, affine=False,
track_running_stats=False)
        )
        (2): ReLU(inplace=True)
      )
    )
    (conv_f2x): Conv2dIBNormRelu(
      (layers): Sequential(
        (0): Conv2d(64, 32, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
        (1): IBNorm(
          (bnorm): BatchNorm2d(16, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (inorm): InstanceNorm2d(16, eps=1e-05, momentum=0.1, affine=False,
track_running_stats=False)
        )
        (2): ReLU(inplace=True)
      )
    )
    (conv_f): Sequential(
      (0): Conv2dIBNormRelu(
        (layers): Sequential(
          (0): Conv2d(35, 16, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
          (1): IBNorm(
            (bnorm): BatchNorm2d(8, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
            (inorm): InstanceNorm2d(8, eps=1e-05, momentum=0.1, affine=False,
track_running_stats=False)
          )
          (2): ReLU(inplace=True)
        )
      )
      (1): Conv2dIBNormRelu(
        (layers): Sequential(
          (0): Conv2d(16, 1, kernel_size=(1, 1), stride=(1, 1))
        )
      )
    )
  )
)
```

[71]:

```python
# =====================================================
#   Part 4 - Define Loss Function and Optimizer
# =====================================================
import torch.nn.functional as F
```

```python
# --- Loss Function ---
# MODNet predicts the alpha matte, so we use L1 (absolute error) loss
# You can combine with L2 or add edge loss later for refinement.
def matte_loss(pred, target):
    """Compute L1 loss between predicted and ground-truth matte."""
    return F.l1_loss(pred, target)

# --- Optimizer ---
# AdamW is preferred for stability; fine-tuning usually needs a small LR
optimizer = torch.optim.AdamW(
    model.parameters(),
    lr=1e-5,          # lower learning rate for fine-tuning
    weight_decay=1e-4
)

# --- Optional: Learning Rate Scheduler ---
scheduler = torch.optim.lr_scheduler.StepLR(optimizer, step_size=5, gamma=0.5)

print("  Optimizer and loss function are ready.")
```

```
 Optimizer and loss function are ready.
```

[72]:

```python
print(f"Train dataset size: {len(train_ds)}")
print(f"Validation dataset size: {len(val_ds)}")
print(f"Train loader size: {len(train_loader)}")
print(f"Validation loader size: {len(val_loader)}")
```

```
Train dataset size: 429
Validation dataset size: 91
Train loader size: 108
Validation loader size: 46
```

[73]:

```python
def train_one_epoch(model, dataloader, optimizer, loss_fn, device):
    model.train()
    running_loss = 0.0
    loop = tqdm(dataloader, desc="Training")

    for imgs, mattes in loop:
        imgs, mattes = imgs.to(device), mattes.to(device)

        optimizer.zero_grad()
        _, _, pred_mattes = model(imgs, True)
        loss = loss_fn(pred_mattes, mattes)
        loss.backward()
        optimizer.step()

        running_loss += loss.item()
        loop.set_postfix(loss=loss.item())
    return running_loss / len(dataloader)
```

[74]:

```python
def validate_one_epoch(model, dataloader, loss_fn, device):
    model.eval()
    val_loss = 0.0

    with torch.no_grad():
        for imgs, mattes in tqdm(dataloader, desc="Validation"):
            imgs, mattes = imgs.to(device), mattes.to(device)
            _, _, pred_mattes = model(imgs, True)
            loss = loss_fn(pred_mattes, mattes)
            val_loss += loss.item()

    return val_loss / len(dataloader)
```

[75]:

```python
def test_model(model, dataloader, device, save_dir=None):
    model.eval()
    os.makedirs(save_dir, exist_ok=True)

    with torch.no_grad():
        for idx, (imgs, _) in enumerate(tqdm(dataloader, desc="Testing")):
            imgs = imgs.to(device)
            _, _, pred_mattes = model(imgs, True)

            for i in range(imgs.size(0)):
                matte_np = pred_mattes[i].squeeze().cpu().numpy() * 255
                matte_img = Image.fromarray(matte_np.astype(np.uint8))
                matte_img.save(f"{save_dir}/matte_{idx}_{i}.png")
```

[90]:

```python
import numpy as np
from skimage.metrics import structural_similarity

def compute_metrics(pred, gt):
    pred_np = pred.squeeze().cpu().numpy()
    gt_np = gt.squeeze().cpu().numpy()

    # Normalize to [0, 1]
    pred_np = np.clip(pred_np, 0, 1)
    gt_np = np.clip(gt_np, 0, 1)

    # SSIM
    ssim_val = structural_similarity(pred_np, gt_np, data_range=1.0)

    # MAE
    mae_val = np.mean(np.abs(pred_np - gt_np))
    # IoU
    pred_bin = pred_np > 0.5
    gt_bin = gt_np > 0.5
    intersection = np.logical_and(pred_bin, gt_bin).sum()
    union = np.logical_or(pred_bin, gt_bin).sum()
    iou_val = intersection / union if union > 0 else 0.0

    return ssim_val, mae_val, iou_val
```

[91]:
```
import torch

def evaluate_metrics(model, val_loader, device):
    ssim_total, mae_total, iou_total, count = 0.0, 0.0, 0.0, 0
    model.eval()
    with torch.no_grad():
        val_iter = iter(val_loader)
        try:
            imgs, mattes = next(val_iter)
            imgs, mattes = imgs.to(device), mattes.to(device)
            _, _, pred_mattes = model(imgs, True)
            for i in range(imgs.size(0)):
                ssim_val, mae_val, iou_val = compute_metrics(pred_mattes[i],␣
 ↪mattes[i])
                ssim_total += ssim_val
                mae_total += mae_val
                iou_total += iou_val
                count += 1
        except StopIteration:
            print("  Validation loader is empty. Skipping metrics.")
        finally:
            del val_iter

    if count > 0:
        return (
            ssim_total / count,
            mae_total / count,
            iou_total / count
        )
    else:
        return None, None, None
```

[92]:
```
EPOCHS = 30
train_losses = []
val_losses = []
ssim_scores = []
mae_scores = []
iou_scores = []
```

```python
for epoch in range(EPOCHS):
    print(f"\n  Epoch {epoch+1}/{EPOCHS}")

    # Training
    train_loss = train_one_epoch(model, train_loader, optimizer, matte_loss,␣
 ↪device)
    train_losses.append(train_loss)

    # Validation
    val_loss = validate_one_epoch(model, val_loader, matte_loss, device)
    val_losses.append(val_loss)

    # Metrics
    ssim_val, mae_val, iou_val = evaluate_metrics(model, val_loader, device)
    ssim_scores.append(ssim_val)
    mae_scores.append(mae_val)
    iou_scores.append(iou_val)

    print(f"  Train Loss: {train_loss:.6f} |   Val Loss: {val_loss:.6f}")
    if ssim_val is not None:
        print(f"  SSIM: {ssim_val:.4f} | MAE: {mae_val:.4f} | IoU: {iou_val:.
 ↪4f}")
    else:
        print("  SSIM, MAE, IoU: Skipped due to empty validation batch.")

    scheduler.step()
```

```
  Epoch 1/30

Training:   0%|          | 0/108 [00:00<?, ?it/s]

Validation:   0%|          | 0/46 [00:00<?, ?it/s]

  Train Loss: 0.016520 |   Val Loss: 0.020528
  SSIM: 0.9855 | MAE: 0.0041 | IoU: 0.9973

  Epoch 2/30

Training:   0%|          | 0/108 [00:00<?, ?it/s]

Validation:   0%|          | 0/46 [00:00<?, ?it/s]

  Train Loss: 0.017457 |   Val Loss: 0.022036
  SSIM: 0.9837 | MAE: 0.0043 | IoU: 0.9970

  Epoch 3/30

Training:   0%|          | 0/108 [00:00<?, ?it/s]
```

```
Exception ignored in: <function _MultiProcessingDataLoaderIter.__del__ at
0x7c5dfe1358a0>
Traceback (most recent call last):
  File "/usr/local/lib/python3.12/dist-packages/torch/utils/data/dataloader.py",
line 1664, in __del__
    self._shutdown_workers()
  File "/usr/local/lib/python3.12/dist-packages/torch/utils/data/dataloader.py",
line 1647, in _shutdown_workers
    if w.is_alive():
       ^^^^^^^^^^^^
  File "/usr/lib/python3.12/multiprocessing/process.py", line 160, in is_alive
    assert self._parent_pid == os.getpid(), 'can only test a child process'
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
AssertionError: can only test a child process
Exception ignored in: <function _MultiProcessingDataLoaderIter.__del__ at
0x7c5dfe1358a0>
Traceback (most recent call last):
  File "/usr/local/lib/python3.12/dist-packages/torch/utils/data/dataloader.py",
line 1664, in __del__
    self._shutdown_workers()
  File "/usr/local/lib/python3.12/dist-packages/torch/utils/data/dataloader.py",
line 1647, in _shutdown_workers
    if w.is_alive():
       ^^^^^^^^^^^^
  File "/usr/lib/python3.12/multiprocessing/process.py", line 160, in is_alive
    assert self._parent_pid == os.getpid(), 'can only test a child process'
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
AssertionError: can only test a child process
Exception ignored in: <function _MultiProcessingDataLoaderIter.__del__ at
0x7c5dfe1358a0>
Traceback (most recent call last):
  File "/usr/local/lib/python3.12/dist-packages/torch/utils/data/dataloader.py",
line 1664, in __del__
    self._shutdown_workers()
  File "/usr/local/lib/python3.12/dist-packages/torch/utils/data/dataloader.py",
line 1647, in _shutdown_workers
    if w.is_alive():
       ^^^^^^^^^^^^
  File "/usr/lib/python3.12/multiprocessing/process.py", line 160, in is_alive
    assert self._parent_pid == os.getpid(), 'can only test a child process'
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
AssertionError: can only test a child process

Validation:   0%|          | 0/46 [00:00<?, ?it/s]

 Train Loss: 0.016165 |   Val Loss: 0.020851
 SSIM: 0.9854 | MAE: 0.0041 | IoU: 0.9972

 Epoch 4/30
```

```
Training:   0%|          | 0/108 [00:00<?, ?it/s]

Validation:   0%|          | 0/46 [00:00<?, ?it/s]

 Train Loss: 0.017021 |  Val Loss: 0.020954
 SSIM: 0.9854 | MAE: 0.0042 | IoU: 0.9972

 Epoch 5/30

Training:   0%|          | 0/108 [00:00<?, ?it/s]

Validation:   0%|          | 0/46 [00:00<?, ?it/s]

 Train Loss: 0.016599 |  Val Loss: 0.021427
 SSIM: 0.9865 | MAE: 0.0039 | IoU: 0.9972

 Epoch 6/30

Training:   0%|          | 0/108 [00:00<?, ?it/s]

Validation:   0%|          | 0/46 [00:00<?, ?it/s]

 Train Loss: 0.016148 |  Val Loss: 0.021268
 SSIM: 0.9852 | MAE: 0.0042 | IoU: 0.9971

 Epoch 7/30

Training:   0%|          | 0/108 [00:00<?, ?it/s]

Validation:   0%|          | 0/46 [00:00<?, ?it/s]

 Train Loss: 0.016012 |  Val Loss: 0.020783
 SSIM: 0.9854 | MAE: 0.0042 | IoU: 0.9973

 Epoch 8/30

Training:   0%|          | 0/108 [00:00<?, ?it/s]

Validation:   0%|          | 0/46 [00:00<?, ?it/s]

 Train Loss: 0.016189 |  Val Loss: 0.022637
 SSIM: 0.9840 | MAE: 0.0042 | IoU: 0.9971

 Epoch 9/30

Training:   0%|          | 0/108 [00:00<?, ?it/s]

Validation:   0%|          | 0/46 [00:00<?, ?it/s]

 Train Loss: 0.018858 |  Val Loss: 0.020297
 SSIM: 0.9849 | MAE: 0.0043 | IoU: 0.9970

 Epoch 10/30

Training:   0%|          | 0/108 [00:00<?, ?it/s]

Validation:   0%|          | 0/46 [00:00<?, ?it/s]
```

```
 Train Loss: 0.015905 |   Val Loss: 0.021655
 SSIM: 0.9847 | MAE: 0.0042 | IoU: 0.9972

 Epoch 11/30

Training:   0%|          | 0/108 [00:00<?, ?it/s]

Validation:   0%|          | 0/46 [00:00<?, ?it/s]

 Train Loss: 0.017001 |   Val Loss: 0.020925
 SSIM: 0.9863 | MAE: 0.0040 | IoU: 0.9973

 Epoch 12/30

Training:   0%|          | 0/108 [00:00<?, ?it/s]

Exception ignored in: <function _MultiProcessingDataLoaderIter.__del__ at
0x7c5dfe1358a0>
Traceback (most recent call last):
  File "/usr/local/lib/python3.12/dist-packages/torch/utils/data/dataloader.py",
line 1664, in __del__
    self._shutdown_workers()
  File "/usr/local/lib/python3.12/dist-packages/torch/utils/data/dataloader.py",
line 1647, in _shutdown_workers
    if w.is_alive():
       ^^^^^^^^^^^^
  File "/usr/lib/python3.12/multiprocessing/process.py", line 160, in is_alive
    assert self._parent_pid == os.getpid(), 'can only test a child process'
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
AssertionError: can only test a child process
Exception ignored in: <function _MultiProcessingDataLoaderIter.__del__ at
0x7c5dfe1358a0>
Traceback (most recent call last):
  File "/usr/local/lib/python3.12/dist-packages/torch/utils/data/dataloader.py",
line 1664, in __del__
    self._shutdown_workers()
  File "/usr/local/lib/python3.12/dist-packages/torch/utils/data/dataloader.py",
line 1647, in _shutdown_workers
    if w.is_alive():
       ^^^^^^^^^^^^
  File "/usr/lib/python3.12/multiprocessing/process.py", line 160, in is_alive
    assert self._parent_pid == os.getpid(), 'can only test a child process'
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
AssertionError: can only test a child process
Exception ignored in: <function _MultiProcessingDataLoaderIter.__del__ at
0x7c5dfe1358a0>
Traceback (most recent call last):
  File "/usr/local/lib/python3.12/dist-packages/torch/utils/data/dataloader.py",
line 1664, in __del__
    self._shutdown_workers()
  File "/usr/local/lib/python3.12/dist-packages/torch/utils/data/dataloader.py",
line 1647, in _shutdown_workers
    if w.is_alive():
       ^^^^^^^^^^^^
  File "/usr/lib/python3.12/multiprocessing/process.py", line 160, in is_alive
    assert self._parent_pid == os.getpid(), 'can only test a child process'
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
AssertionError: can only test a child process
Exception ignored in: <function _MultiProcessingDataLoaderIter.__del__ at
0x7c5dfe1358a0>
Traceback (most recent call last):
  File "/usr/local/lib/python3.12/dist-packages/torch/utils/data/dataloader.py",
line 1664, in __del__
    self._shutdown_workers()
  File "/usr/local/lib/python3.12/dist-packages/torch/utils/data/dataloader.py",
line 1647, in _shutdown_workers
    if w.is_alive():
       ^^^^^^^^^^^^
  File "/usr/lib/python3.12/multiprocessing/process.py", line 160, in is_alive
    assert self._parent_pid == os.getpid(), 'can only test a child process'
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
AssertionError: can only test a child process
Exception ignored in: <function _MultiProcessingDataLoaderIter.__del__ at
0x7c5dfe1358a0>
Traceback (most recent call last):
  File "/usr/local/lib/python3.12/dist-packages/torch/utils/data/dataloader.py",
line 1664, in __del__
    self._shutdown_workers()
  File "/usr/local/lib/python3.12/dist-packages/torch/utils/data/dataloader.py",
line 1647, in _shutdown_workers
    if w.is_alive():
       ^^^^^^^^^^^^
  File "/usr/lib/python3.12/multiprocessing/process.py", line 160, in is_alive
    assert self._parent_pid == os.getpid(), 'can only test a child process'
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
AssertionError: can only test a child process
Exception ignored in: <function _MultiProcessingDataLoaderIter.__del__ at
0x7c5dfe1358a0>
Traceback (most recent call last):
  File "/usr/local/lib/python3.12/dist-packages/torch/utils/data/dataloader.py",
line 1664, in __del__
    self._shutdown_workers()
  File "/usr/local/lib/python3.12/dist-packages/torch/utils/data/dataloader.py",
line 1647, in _shutdown_workers

if w.is_alive():       ^^^^^^^^^^^^
  File "/usr/lib/python3.12/multiprocessing/process.py", line 160, in is_alive
    assert self._parent_pid == os.getpid(), 'can only test a child process'
```

```
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
AssertionError: can only test a child process
Exception ignored in: <function _MultiProcessingDataLoaderIter.__del__ at
0x7c5dfe1358a0>
Traceback (most recent call last):
  File "/usr/local/lib/python3.12/dist-packages/torch/utils/data/dataloader.py",
line 1664, in __del__
    self._shutdown_workers()
  File "/usr/local/lib/python3.12/dist-packages/torch/utils/data/dataloader.py",
line 1647, in _shutdown_workers
    if w.is_alive():
       ^^^^^^^^^^^^
  File "/usr/lib/python3.12/multiprocessing/process.py", line 160, in is_alive
    assert self._parent_pid == os.getpid(), 'can only test a child process'
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
AssertionError: can only test a child process
Exception ignored in: <function _MultiProcessingDataLoaderIter.__del__ at
0x7c5dfe1358a0>
Traceback (most recent call last):
  File "/usr/local/lib/python3.12/dist-packages/torch/utils/data/dataloader.py",
line 1664, in __del__
    self._shutdown_workers()
  File "/usr/local/lib/python3.12/dist-packages/torch/utils/data/dataloader.py",
line 1647, in _shutdown_workers
    if w.is_alive():
       ^^^^^^^^^^^^
  File "/usr/lib/python3.12/multiprocessing/process.py", line 160, in is_alive
    assert self._parent_pid == os.getpid(), 'can only test a child process'
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
AssertionError: can only test a child process

Validation:   0%|          | 0/46 [00:00<?, ?it/s]

 Train Loss: 0.016204 |   Val Loss: 0.022491
 SSIM: 0.9867 | MAE: 0.0040 | IoU: 0.9973


 Epoch 13/30

Training:   0%|          | 0/108 [00:00<?, ?it/s]

Validation:   0%|          | 0/46 [00:00<?, ?it/s]

 Train Loss: 0.015766 |   Val Loss: 0.021137
 SSIM: 0.9858 | MAE: 0.0040 | IoU: 0.9974


 Epoch 14/30

Training:   0%|          | 0/108 [00:00<?, ?it/s]

Validation:   0%|          | 0/46 [00:00<?, ?it/s]
```

```
 Train Loss: 0.015740 |  Val Loss: 0.021789
 SSIM: 0.9864 | MAE: 0.0040 | IoU: 0.9971

 Epoch 15/30
Training:   0%|          | 0/108 [00:00<?, ?it/s]
Validation:   0%|          | 0/46 [00:00<?, ?it/s]
 Train Loss: 0.014932 |  Val Loss: 0.021045
 SSIM: 0.9857 | MAE: 0.0039 | IoU: 0.9974

 Epoch 16/30
Training:   0%|          | 0/108 [00:00<?, ?it/s]
Validation:   0%|          | 0/46 [00:00<?, ?it/s]
 Train Loss: 0.015973 |  Val Loss: 0.023023
 SSIM: 0.9869 | MAE: 0.0040 | IoU: 0.9972

 Epoch 17/30
Training:   0%|          | 0/108 [00:00<?, ?it/s]
Validation:   0%|          | 0/46 [00:00<?, ?it/s]
 Train Loss: 0.015631 |  Val Loss: 0.022792
 SSIM: 0.9863 | MAE: 0.0039 | IoU: 0.9972

 Epoch 18/30
Training:   0%|          | 0/108 [00:00<?, ?it/s]
Validation:   0%|          | 0/46 [00:00<?, ?it/s]
 Train Loss: 0.016292 |  Val Loss: 0.022069
 SSIM: 0.9860 | MAE: 0.0039 | IoU: 0.9970

 Epoch 19/30
Training:   0%|          | 0/108 [00:00<?, ?it/s]
Validation:   0%|          | 0/46 [00:00<?, ?it/s]
 Train Loss: 0.016151 |  Val Loss: 0.021422
 SSIM: 0.9850 | MAE: 0.0040 | IoU: 0.9973

 Epoch 20/30
Training:   0%|          | 0/108 [00:00<?, ?it/s]
Validation:   0%|          | 0/46 [00:00<?, ?it/s]
 Train Loss: 0.016279 |  Val Loss: 0.021753
 SSIM: 0.9850 | MAE: 0.0041 | IoU: 0.9972
 Epoch 21/30

Training:   0%|          | 0/108 [00:00<?, ?it/s]

Validation:   0%|          | 0/46 [00:00<?, ?it/s]

 Train Loss: 0.015088 |  Val Loss: 0.022201
 SSIM: 0.9860 | MAE: 0.0040 | IoU: 0.9973

 Epoch 22/30

Training:   0%|          | 0/108 [00:00<?, ?it/s]

Validation:   0%|          | 0/46 [00:00<?, ?it/s]

 Train Loss: 0.016567 |  Val Loss: 0.022741
 SSIM: 0.9865 | MAE: 0.0039 | IoU: 0.9974

 Epoch 23/30

Training:   0%|          | 0/108 [00:00<?, ?it/s]

Validation:   0%|          | 0/46 [00:00<?, ?it/s]

 Train Loss: 0.015748 |  Val Loss: 0.021997
 SSIM: 0.9860 | MAE: 0.0039 | IoU: 0.9971

 Epoch 24/30

Training:   0%|          | 0/108 [00:00<?, ?it/s]

Validation:   0%|          | 0/46 [00:00<?, ?it/s]

 Train Loss: 0.015340 |  Val Loss: 0.020766
 SSIM: 0.9862 | MAE: 0.0039 | IoU: 0.9972

 Epoch 25/30

Training:   0%|          | 0/108 [00:00<?, ?it/s]

Validation:   0%|          | 0/46 [00:00<?, ?it/s]

 Train Loss: 0.015957 |  Val Loss: 0.021396
 SSIM: 0.9879 | MAE: 0.0038 | IoU: 0.9973

 Epoch 26/30

Training:   0%|          | 0/108 [00:00<?, ?it/s]

Exception ignored in: <function _MultiProcessingDataLoaderIter.__del__ at
0x7c5dfe1358a0>
Traceback (most recent call last):
  File "/usr/local/lib/python3.12/dist-packages/torch/utils/data/dataloader.py",
line 1664, in __del__
    self._shutdown_workers()
  File "/usr/local/lib/python3.12/dist-packages/torch/utils/data/dataloader.py",
line 1647, in _shutdown_workers
    if w.is_alive():
       ^^^^^^^^^^^^
  File "/usr/lib/python3.12/multiprocessing/process.py", line 160, in is_alive
    assert self._parent_pid == os.getpid(), 'can only test a child process'
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
AssertionError: can only test a child process
Exception ignored in: <function _MultiProcessingDataLoaderIter.__del__ at
0x7c5dfe1358a0>
Traceback (most recent call last):
  File "/usr/local/lib/python3.12/dist-packages/torch/utils/data/dataloader.py",
line 1664, in __del__
    self._shutdown_workers()
  File "/usr/local/lib/python3.12/dist-packages/torch/utils/data/dataloader.py",
line 1647, in _shutdown_workers
    if w.is_alive():
       ^^^^^^^^^^^^
  File "/usr/lib/python3.12/multiprocessing/process.py", line 160, in is_alive
    assert self._parent_pid == os.getpid(), 'can only test a child process'
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
AssertionError: can only test a child process
Exception ignored in: <function _MultiProcessingDataLoaderIter.__del__ at
0x7c5dfe1358a0>
Traceback (most recent call last):
  File "/usr/local/lib/python3.12/dist-packages/torch/utils/data/dataloader.py",
line 1664, in __del__
    self._shutdown_workers()
  File "/usr/local/lib/python3.12/dist-packages/torch/utils/data/dataloader.py",
line 1647, in _shutdown_workers
    if w.is_alive():
       ^^^^^^^^^^^^
  File "/usr/lib/python3.12/multiprocessing/process.py", line 160, in is_alive
    assert self._parent_pid == os.getpid(), 'can only test a child process'
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
AssertionError: can only test a child process
Exception ignored in: <function _MultiProcessingDataLoaderIter.__del__ at
0x7c5dfe1358a0>
Traceback (most recent call last):
  File "/usr/local/lib/python3.12/dist-packages/torch/utils/data/dataloader.py",
line 1664, in __del__
    self._shutdown_workers()
  File "/usr/local/lib/python3.12/dist-packages/torch/utils/data/dataloader.py",
line 1647, in _shutdown_workers
    if w.is_alive():
       ^^^^^^^^^^^^
  File "/usr/lib/python3.12/multiprocessing/process.py", line 160, in is_alive
    assert self._parent_pid == os.getpid(), 'can only test a child process'
```

```
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
AssertionError: can only test a child process
Exception ignored in: <function _MultiProcessingDataLoaderIter.__del__ at
0x7c5dfe1358a0>
Traceback (most recent call last):
  File "/usr/local/lib/python3.12/dist-packages/torch/utils/data/dataloader.py",
line 1664, in __del__
    self._shutdown_workers()
  File "/usr/local/lib/python3.12/dist-packages/torch/utils/data/dataloader.py",
line 1647, in _shutdown_workers
    if w.is_alive():
       ^^^^^^^^^^^^
  File "/usr/lib/python3.12/multiprocessing/process.py", line 160, in is_alive
    assert self._parent_pid == os.getpid(), 'can only test a child process'
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
AssertionError: can only test a child process
Exception ignored in: <function _MultiProcessingDataLoaderIter.__del__ at
0x7c5dfe1358a0>
Traceback (most recent call last):
  File "/usr/local/lib/python3.12/dist-packages/torch/utils/data/dataloader.py",
line 1664, in __del__
    self._shutdown_workers()
  File "/usr/local/lib/python3.12/dist-packages/torch/utils/data/dataloader.py",
line 1647, in _shutdown_workers
    if w.is_alive():
       ^^^^^^^^^^^^
  File "/usr/lib/python3.12/multiprocessing/process.py", line 160, in is_alive
    assert self._parent_pid == os.getpid(), 'can only test a child process'
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
AssertionError:
can only test a child processException ignored in: <function
_MultiProcessingDataLoaderIter.__del__ at 0x7c5dfe1358a0>
Traceback (most recent call last):
  File "/usr/local/lib/python3.12/dist-packages/torch/utils/data/dataloader.py",
line 1664, in __del__
    self._shutdown_workers()
  File "/usr/local/lib/python3.12/dist-packages/torch/utils/data/dataloader.py",
line 1647, in _shutdown_workers
    if w.is_alive():
       ^^^^^^^^^^^^
  File "/usr/lib/python3.12/multiprocessing/process.py", line 160, in is_alive
    assert self._parent_pid == os.getpid(), 'can only test a child process'
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
AssertionError: can only test a child process
Exception ignored in: <function _MultiProcessingDataLoaderIter.__del__ at
0x7c5dfe1358a0>
Traceback (most recent call last):
  File "/usr/local/lib/python3.12/dist-packages/torch/utils/data/dataloader.py",
line 1664, in __del__
    self._shutdown_workers()
  File "/usr/local/lib/python3.12/dist-packages/torch/utils/data/dataloader.py",
line 1647, in _shutdown_workers
    if w.is_alive():
       ^^^^^^^^^^^^
  File "/usr/lib/python3.12/multiprocessing/process.py", line 160, in is_alive
    assert self._parent_pid == os.getpid(), 'can only test a child process'
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
AssertionError: can only test a child process
```

```
Validation:   0%|          | 0/46 [00:00<?, ?it/s]
```

```
 Train Loss: 0.016543 |  Val Loss: 0.023501
 SSIM: 0.9855 | MAE: 0.0040 | IoU: 0.9972

 Epoch 27/30
```

```
Training:   0%|          | 0/108 [00:00<?, ?it/s]
```

```
Validation:   0%|          | 0/46 [00:00<?, ?it/s]
```

```
 Train Loss: 0.015815 |  Val Loss: 0.022160
 SSIM: 0.9876 | MAE: 0.0038 | IoU: 0.9973

 Epoch 28/30
```

```
Training:   0%|          | 0/108 [00:00<?, ?it/s]
```

```
Validation:   0%|          | 0/46 [00:00<?, ?it/s]
```

```
 Train Loss: 0.015696 |  Val Loss: 0.021147
 SSIM: 0.9856 | MAE: 0.0041 | IoU: 0.9972

 Epoch 29/30
```

```
Training:   0%|          | 0/108 [00:00<?, ?it/s]
```

```
Validation:   0%|          | 0/46 [00:00<?, ?it/s]
```

```
 Train Loss: 0.015033 |  Val Loss: 0.021802
 SSIM: 0.9855 | MAE: 0.0040 | IoU: 0.9972

 Epoch 30/30
```

```
Training:   0%|          | 0/108 [00:00<?, ?it/s]
```

```
Validation:   0%|          | 0/46 [00:00<?, ?it/s]
```

```
 Train Loss: 0.016785 |  Val Loss: 0.021115
 SSIM: 0.9872 | MAE: 0.0039 | IoU: 0.9975
```

[99]:

```python
print("Lengths:")
print("Train Losses:", len(train_losses))
print("Val Losses:", len(val_losses))
print("SSIM:", len(ssim_scores))
print("MAE:", len(mae_scores))
print("IoU:", len(iou_scores))
```

```
Lengths:
Train Losses: 30
Val Losses: 30
SSIM: 30
MAE: 30
IoU: 30
```

[100]:

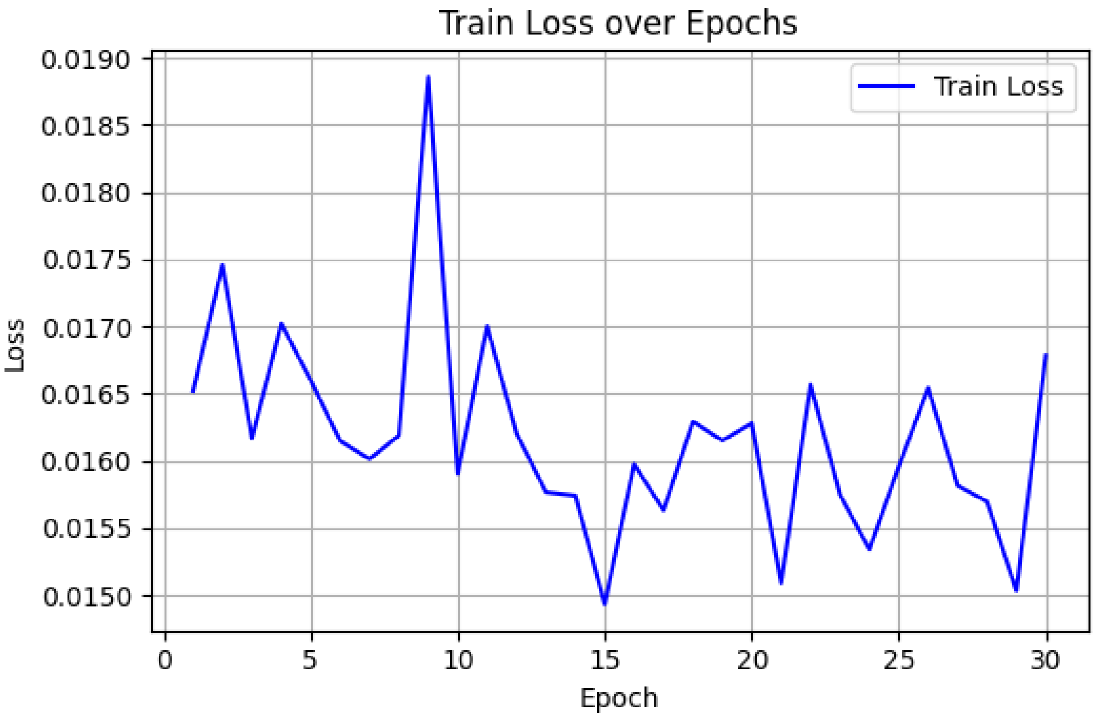
```
import math

filtered = [
    (t, v, s, m, i)
    for t, v, s, m, i in zip(train_losses, val_losses, ssim_scores, mae_scores,␣
 ↪iou_scores)
    if s is not None and not math.isnan(s)
]

train_losses, val_losses, ssim_scores, mae_scores, iou_scores = zip(*filtered)
```

[103]:

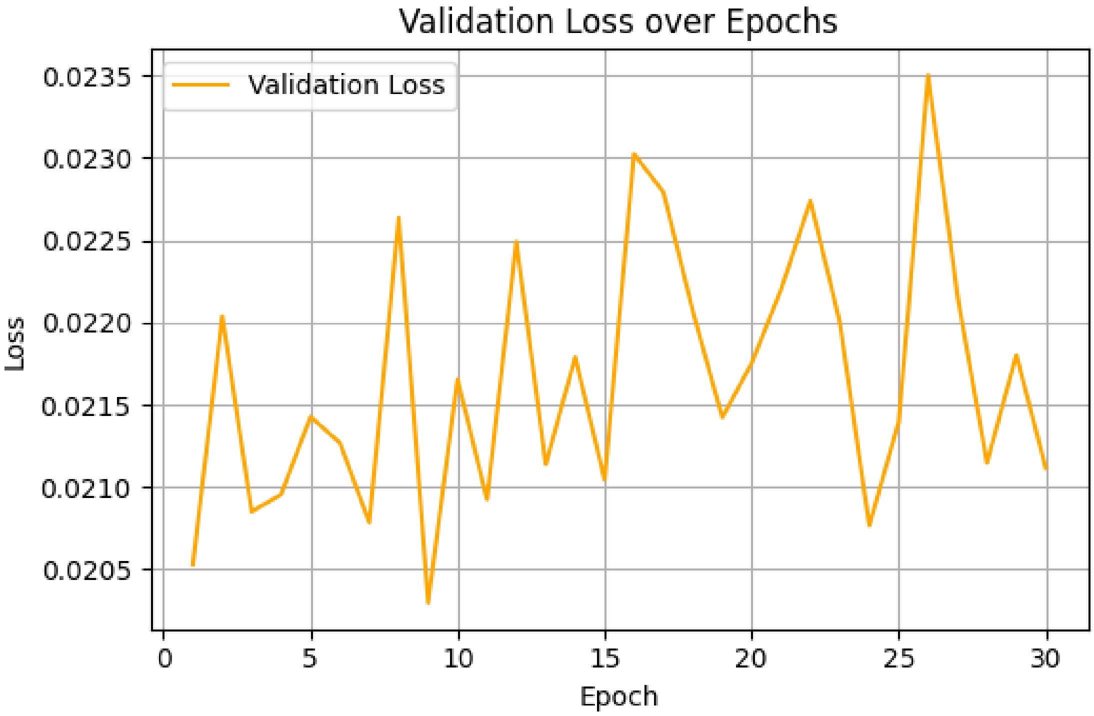
```
import matplotlib.pyplot as plt

def plot_metric(metric_values, title, ylabel, color="blue"):
    epochs = range(1, len(metric_values) + 1)
    plt.figure(figsize=(6, 4))
    plt.plot(epochs, metric_values, label=title, color=color)
    plt.title(f"{title} over Epochs")
    plt.xlabel("Epoch")
    plt.ylabel(ylabel)
    plt.legend()
    plt.grid(True)
    plt.tight_layout()
    plt.show()
```

[105]:

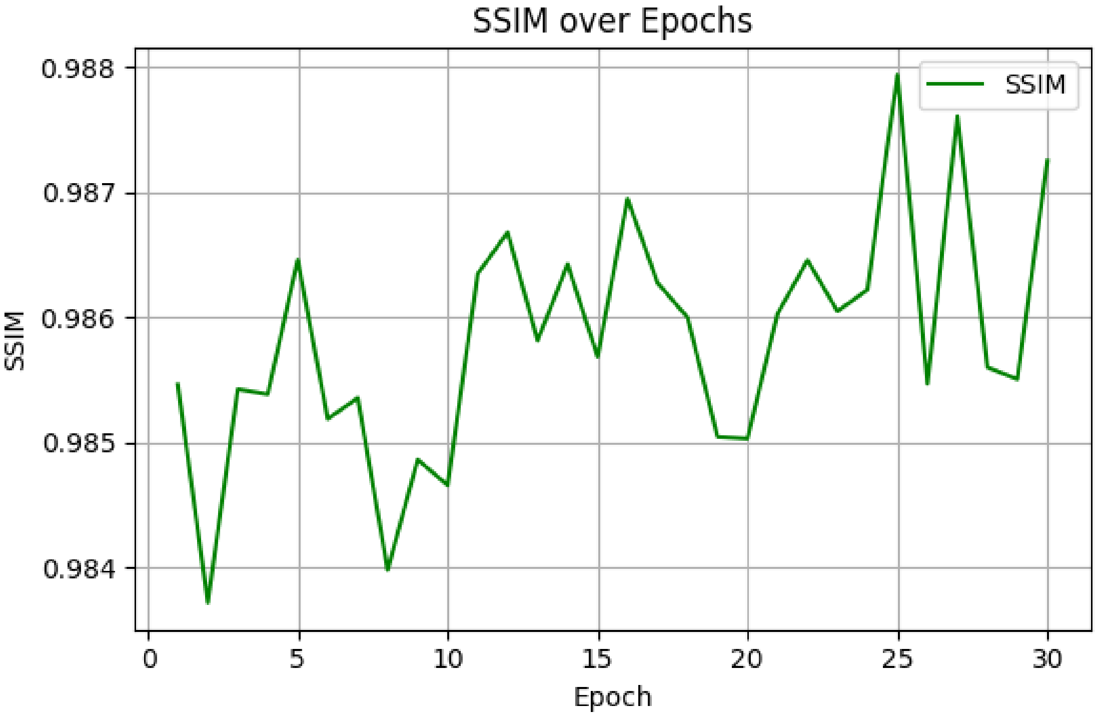
```
plot_metric(train_losses, "Train Loss", "Loss", "blue")
```



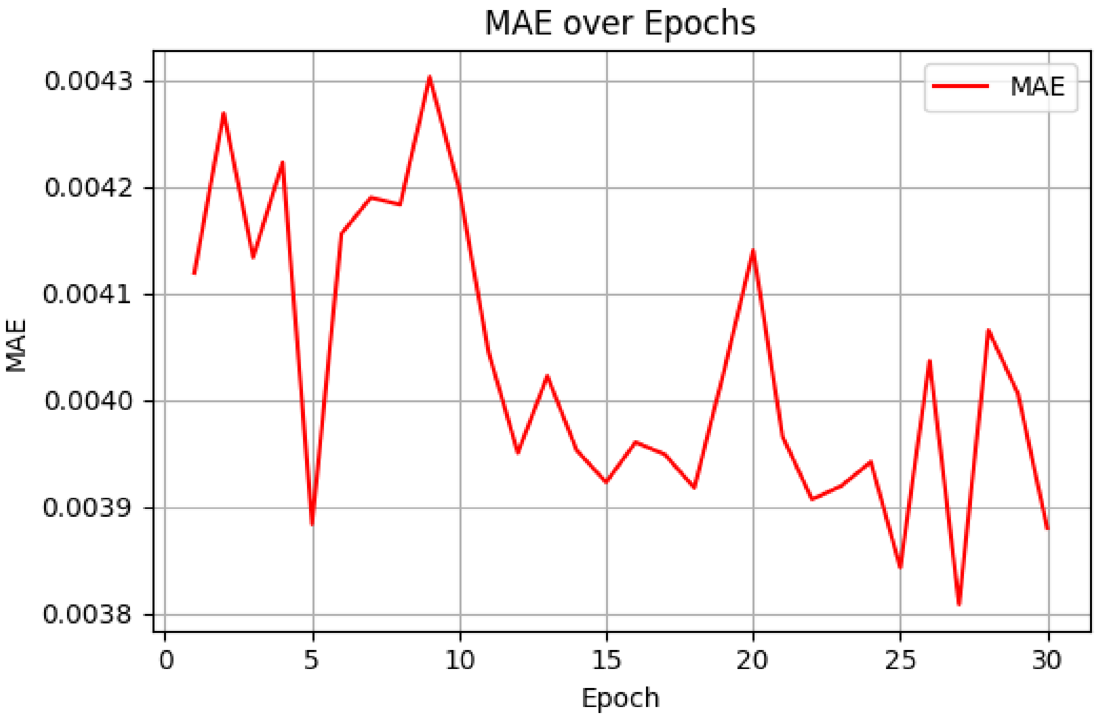
```
[106]: plot_metric(val_losses, "Validation Loss", "Loss", "orange")
```



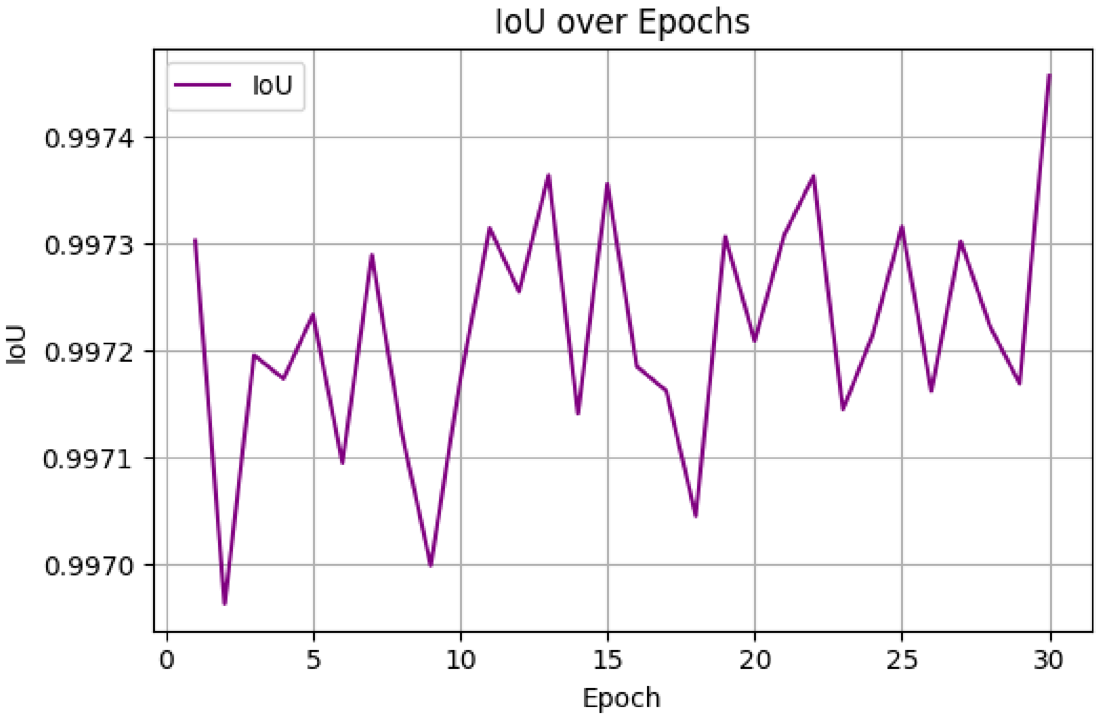
[107]:
```
plot_metric(ssim_scores, "SSIM", "SSIM", "green")
```



[108]:
```
plot_metric(mae_scores, "MAE", "MAE", "red")
```



```
[109]: plot_metric(iou_scores, "IoU", "IoU", "purple")
```



[110]:

```
SAVE_DIR = Path('/content/drive/MyDrive/modnet_data/output_models')
SAVE_DIR.mkdir(parents=True, exist_ok=True)

MODEL_PATH = SAVE_DIR / 'modnet_finetuned_photographic.ckpt'
torch.save(model.state_dict(), MODEL_PATH)
print(f" Model saved to: {MODEL_PATH}")
```

```
 Model saved to: /content/drive/MyDrive/modnet_data/output_models/modnet_finetu
ned_photographic.ckpt
```

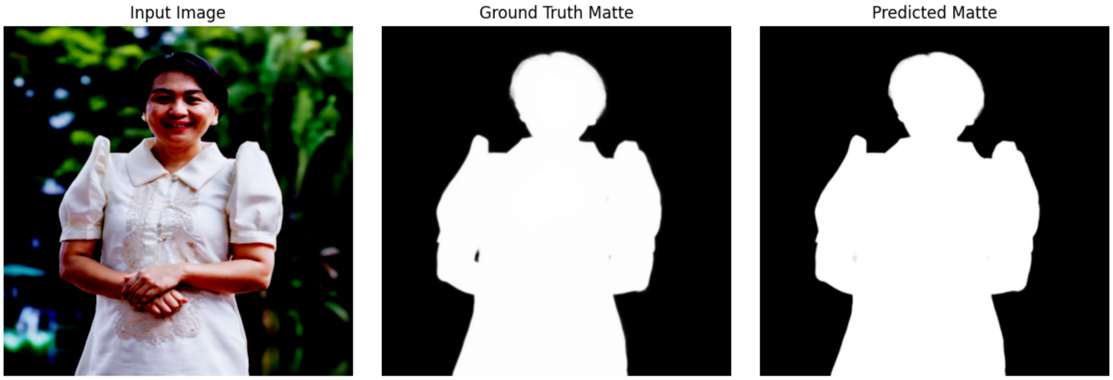
[111]:

```
import matplotlib.pyplot as plt
import torch

def plot_val_samples(model, val_loader, device, num_samples=3):
    model.eval()
    with torch.no_grad():
        val_iter = iter(val_loader)
        try:
            imgs, mattes = next(val_iter)
            imgs, mattes = imgs.to(device), mattes.to(device)
            _, _, pred_mattes = model(imgs, True)

            for i in range(min(num_samples, imgs.size(0))):
                fig, axs = plt.subplots(1, 3, figsize=(12, 4))
                axs[0].imshow(imgs[i].permute(1, 2, 0).cpu().numpy())
                axs[0].set_title("Input Image")
                axs[0].axis("off")

                axs[1].imshow(mattes[i].squeeze().cpu().numpy(), cmap="gray")
                axs[1].set_title("Ground Truth Matte")
                axs[1].axis("off")

                axs[2].imshow(pred_mattes[i].squeeze().cpu().numpy(),␣
 ↪cmap="gray")
                axs[2].set_title("Predicted Matte")
                axs[2].axis("off")

                plt.tight_layout()
                plt.show()

        except StopIteration:
            print(" Validation loader is empty. No samples to plot.")
        finally:
            del val_iter
```

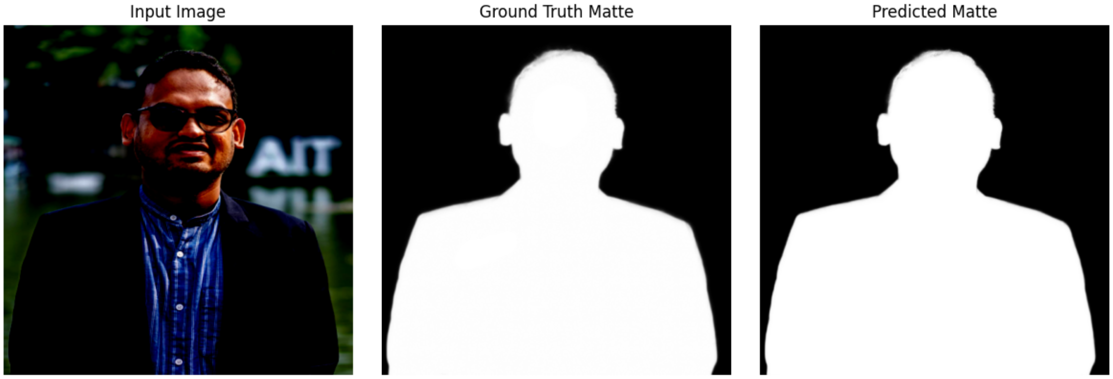
[112]:

```
plot_val_samples(model, val_loader, device, num_samples=3)
```

```
WARNING:matplotlib.image:Clipping input data to the valid range for imshow with
RGB data ([0..1] for floats or [0..255] for integers). Got range [-1.0..1.0].
```



```
WARNING:matplotlib.image:Clipping input data to the valid range for imshow with
RGB data ([0..1] for floats or [0..255] for integers). Got range [-1.0..1.0].
```



[ ]:

```
```