

Colab Setup

```
In [1]: import sys  
  
IS_COLAB = 'google.colab' in sys.modules  
print(f"Running in Google Colab: {IS_COLAB}")
```

Running in Google Colab: True

```
In [2]: import platform  
import psutil  
import subprocess  
import os  
  
if IS_COLAB:  
    print("Google Colab Environment Specifications:")  
    print("="*50)  
  
    # Get system info  
  
    print(f"Operating System: {platform.system()} {platform.release()}")  
    print(f"Architecture: {platform.machine()}")  
    print(f"Python Version: {platform.python_version()}")  
  
    # Memory info  
    memory = psutil.virtual_memory()  
    print(f"Total RAM: {memory.total / (1024**3):.1f} GB")  
    print(f"Available RAM: {memory.available / (1024**3):.1f} GB")  
  
    # CPU info  
    print(f"CPU Cores: {psutil.cpu_count(logical=False)} physical, {psutil.cpu_count(logical=True)} logical")  
  
    # GPU info  
    try:  
        result = subprocess.run(['nvidia-smi', '--query-gpu=name,memory.total', '--format=csv,noheader,nounits'],  
                               capture_output=True, text=True)  
        if result.returncode == 0:  
            gpu_info = result.stdout.strip().split('\n')  
            for i, gpu in enumerate(gpu_info):
```

```
        name, memory = gpu.split(',')
        print(f"GPU {i}: {name}, {memory} MB VRAM")
    else:
        print("GPU: Not detected or nvidia-smi unavailable")
except:
    print("GPU: Not detected")

# Disk space
disk = psutil.disk_usage('/')
print(f"Disk Space: {disk.free / (1024**3):.1f} GB free / {disk.total / (1024**3):.1f} GB total")

print("*"*50)

if not os.path.exists('/content/aai521_3proj'):
    print("WARNING: Cloning project repository required.")
    print("*"*50)
else:
    print("Not running in Google Colab environment")
```

Google Colab Environment Specifications:

```
=====
Operating System: Linux 6.6.105+
Architecture: x86_64
Python Version: 3.12.12
Total RAM: 83.5 GB
Available RAM: 76.3 GB
CPU Cores: 6 physical, 12 logical
GPU 0: NVIDIA A100-SXM4-40GB, 40960 MB VRAM
Disk Space: 78.0 GB free / 235.7 GB total
=====
```

In [3]:

```
import os
import sys

if IS_COLAB:
    print("Running in Google Colab environment.")
    if os.path.exists('/content/aai521_3proj'):
        print("Repository already exists. Pulling latest changes...")
        %cd /content/aai521_3proj
        !git pull
    else:
        print("Cloning repository...")
```

```
!git clone https://github.com/swapnilprakashpatil/aai521_3proj.git
%cd aai521_3proj
%pip install -r requirements.txt
sys.path.append('/content/aai521_3proj/src')
%ls
else:
    print("Running in local environment. Installing packages...")
    %pip install -r ./requirements.txt
    sys.path.append('../src')
```

```
Running in Google Colab environment.  
Repository already exists. Pulling latest changes...  
/content/aa1521_3proj  
Already up to date.  
Already up to date.  
Requirement already satisfied: numpy>=1.24.0 in /usr/local/lib/python3.12/dist-packages (from -r requirements.txt (line 2)) (2.0.2)  
Requirement already satisfied: pandas>=2.0.0 in /usr/local/lib/python3.12/dist-packages (from -r requirements.txt (line 3)) (2.2.2)  
Requirement already satisfied: scikit-learn>=1.3.0 in /usr/local/lib/python3.12/dist-packages (from -r requirements.txt (line 4)) (1.6.1)  
Requirement already satisfied: scipy>=1.11.0 in /usr/local/lib/python3.12/dist-packages (from -r requirements.txt (line 5)) (1.16.3)  
Requirement already satisfied: torch>=2.0.0 in /usr/local/lib/python3.12/dist-packages (from -r requirements.txt (line 8)) (2.9.0+cu126)  
Requirement already satisfied: torchvision>=0.15.0 in /usr/local/lib/python3.12/dist-packages (from -r requirements.txt (line 9)) (0.24.0+cu126)  
Requirement already satisfied: segmentation-models-pytorch>=0.3.3 in /usr/local/lib/python3.12/dist-packages (from -r requirements.txt (line 10)) (0.5.0)  
Requirement already satisfied: albumentations>=1.3.1 in /usr/local/lib/python3.12/dist-packages (from -r requirements.txt (line 11)) (2.0.8)  
Requirement already satisfied: opencv-python>=4.8.0 in /usr/local/lib/python3.12/dist-packages (from -r requirements.txt (line 14)) (4.12.0.88)  
Requirement already satisfied: scikit-image>=0.21.0 in /usr/local/lib/python3.12/dist-packages (from -r requirements.txt (line 15)) (0.25.2)  
Requirement already satisfied: Pillow>=10.0.0 in /usr/local/lib/python3.12/dist-packages (from -r requirements.txt (line 16)) (11.3.0)  
Requirement already satisfied: rasterio>=1.3.8 in /usr/local/lib/python3.12/dist-packages (from -r requirements.txt (line 19)) (1.4.3)  
Requirement already satisfied: geopandas>=0.13.0 in /usr/local/lib/python3.12/dist-packages (from -r requirements.txt (line 20)) (1.1.1)  
Requirement already satisfied: shapely>=2.0.0 in /usr/local/lib/python3.12/dist-packages (from -r requirements.txt (line 21)) (2.1.2)  
Requirement already satisfied: matplotlib>=3.7.0 in /usr/local/lib/python3.12/dist-packages (from -r requirements.txt (line 24)) (3.10.0)  
Requirement already satisfied: seaborn>=0.12.0 in /usr/local/lib/python3.12/dist-packages (from -r requirements.txt (line 25)) (0.13.2)  
Requirement already satisfied: tqdm>=4.65.0 in /usr/local/lib/python3.12/dist-packages (from -r requirements.txt (line 28)) (4.67.1)  
Requirement already satisfied: tensorboard>=2.13.0 in /usr/local/lib/python3.12/dist-packages (from -r requirements.txt (line 31)) (2.19.0)  
Requirement already satisfied: jupyter>=1.0.0 in /usr/local/lib/python3.12/dist-packages (from -r requirements.txt (line 32)) (1.0.0)
```

```
ine 34)) (1.1.1)
Requirement already satisfied: ipykernel>=6.25.0 in /usr/local/lib/python3.12/dist-packages (from -r requirements.txt
(line 35)) (7.1.0)
Requirement already satisfied: ipywidgets>=8.1.0 in /usr/local/lib/python3.12/dist-packages (from -r requirements.txt
(line 36)) (8.1.8)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.12/dist-packages (from pandas>=2.0.0->
-r requirements.txt (line 3)) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.12/dist-packages (from pandas>=2.0.0->-r require
ments.txt (line 3)) (2025.2)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.12/dist-packages (from pandas>=2.0.0->-r requi
rements.txt (line 3)) (2025.2)
Requirement already satisfied: joblib>=1.2.0 in /usr/local/lib/python3.12/dist-packages (from scikit-learn>=1.3.0->-r
requirements.txt (line 4)) (1.5.2)
Requirement already satisfied: threadpoolctl>=3.1.0 in /usr/local/lib/python3.12/dist-packages (from scikit-learn>=1.
3.0->-r requirements.txt (line 4)) (3.6.0)
Requirement already satisfied: numpy>=1.24.0 in /usr/local/lib/python3.12/dist-packages (from -r requirements.txt (li
ne 2)) (2.0.2)
Requirement already satisfied: pandas>=2.0.0 in /usr/local/lib/python3.12/dist-packages (from -r requirements.txt (li
ne 3)) (2.2.2)
Requirement already satisfied: scikit-learn>=1.3.0 in /usr/local/lib/python3.12/dist-packages (from -r requirements.t
xt (line 4)) (1.6.1)
Requirement already satisfied: scipy>=1.11.0 in /usr/local/lib/python3.12/dist-packages (from -r requirements.txt (li
ne 5)) (1.16.3)
Requirement already satisfied: torch>=2.0.0 in /usr/local/lib/python3.12/dist-packages (from -r requirements.txt (lin
e 8)) (2.9.0+cu126)
Requirement already satisfied: torchvision>=0.15.0 in /usr/local/lib/python3.12/dist-packages (from -r requirements.t
xt (line 9)) (0.24.0+cu126)
Requirement already satisfied: segmentation-models-pytorch>=0.3.3 in /usr/local/lib/python3.12/dist-packages (from -r
requirements.txt (line 10)) (0.5.0)
Requirement already satisfied: albumentations>=1.3.1 in /usr/local/lib/python3.12/dist-packages (from -r requirement
s.txt (line 11)) (2.0.8)
Requirement already satisfied: opencv-python>=4.8.0 in /usr/local/lib/python3.12/dist-packages (from -r requirements.
txt (line 14)) (4.12.0.88)
Requirement already satisfied: scikit-image>=0.21.0 in /usr/local/lib/python3.12/dist-packages (from -r requirements.
txt (line 15)) (0.25.2)
Requirement already satisfied: Pillow>=10.0.0 in /usr/local/lib/python3.12/dist-packages (from -r requirements.txt (l
ine 16)) (11.3.0)
Requirement already satisfied: rasterio>=1.3.8 in /usr/local/lib/python3.12/dist-packages (from -r requirements.txt
(line 19)) (1.4.3)
Requirement already satisfied: geopandas>=0.13.0 in /usr/local/lib/python3.12/dist-packages (from -r requirements.txt
(line 20)) (1.1.1)
Requirement already satisfied: shapely>=2.0.0 in /usr/local/lib/python3.12/dist-packages (from -r requirements.txt (1
```

```
ine 21)) (2.1.2)
Requirement already satisfied: matplotlib>=3.7.0 in /usr/local/lib/python3.12/dist-packages (from -r requirements.txt
(line 24)) (3.10.0)
Requirement already satisfied: seaborn>=0.12.0 in /usr/local/lib/python3.12/dist-packages (from -r requirements.txt
(line 25)) (0.13.2)
Requirement already satisfied: tqdm>=4.65.0 in /usr/local/lib/python3.12/dist-packages (from -r requirements.txt (lin
e 28)) (4.67.1)
Requirement already satisfied: tensorboard>=2.13.0 in /usr/local/lib/python3.12/dist-packages (from -r requirements.t
xt (line 31)) (2.19.0)
Requirement already satisfied: jupyter>=1.0.0 in /usr/local/lib/python3.12/dist-packages (from -r requirements.txt (l
ine 34)) (1.1.1)
Requirement already satisfied: ipykernel>=6.25.0 in /usr/local/lib/python3.12/dist-packages (from -r requirements.txt
(line 35)) (7.1.0)
Requirement already satisfied: ipywidgets>=8.1.0 in /usr/local/lib/python3.12/dist-packages (from -r requirements.txt
(line 36)) (8.1.8)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.12/dist-packages (from pandas>=2.0.0-
>-r requirements.txt (line 3)) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.12/dist-packages (from pandas>=2.0.0->-r requir
ements.txt (line 3)) (2025.2)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.12/dist-packages (from pandas>=2.0.0->-r requ
irements.txt (line 3)) (2025.2)
Requirement already satisfied: joblib>=1.2.0 in /usr/local/lib/python3.12/dist-packages (from scikit-learn>=1.3.0->-r
requirements.txt (line 4)) (1.5.2)
Requirement already satisfied: threadpoolctl>=3.1.0 in /usr/local/lib/python3.12/dist-packages (from scikit-learn>=1.
3.0->-r requirements.txt (line 4)) (3.6.0)
Requirement already satisfied: filelock in /usr/local/lib/python3.12/dist-packages (from torch>=2.0.0->-r requirement
s.txt (line 8)) (3.20.0)
Requirement already satisfied: typing-extensions>=4.10.0 in /usr/local/lib/python3.12/dist-packages (from torch>=2.0.
0->-r requirements.txt (line 8)) (4.15.0)
Requirement already satisfied: setuptools in /usr/local/lib/python3.12/dist-packages (from torch>=2.0.0->-r requireme
nts.txt (line 8)) (75.2.0)
Requirement already satisfied: sympy>=1.13.3 in /usr/local/lib/python3.12/dist-packages (from torch>=2.0.0->-r requir
ements.txt (line 8)) (1.14.0)
Requirement already satisfied: networkx>=2.5.1 in /usr/local/lib/python3.12/dist-packages (from torch>=2.0.0->-r requ
irements.txt (line 8)) (3.6)
Requirement already satisfied: jinja2 in /usr/local/lib/python3.12/dist-packages (from torch>=2.0.0->-r requirements.
txt (line 8)) (3.1.6)
Requirement already satisfied: fsspec>=0.8.5 in /usr/local/lib/python3.12/dist-packages (from torch>=2.0.0->-r requir
ements.txt (line 8)) (2025.3.0)
Requirement already satisfied: nvidia-cuda-nvrtc-cu12==12.6.77 in /usr/local/lib/python3.12/dist-packages (from torch
>=2.0.0->-r requirements.txt (line 8)) (12.6.77)
Requirement already satisfied: nvidia-cuda-runtime-cu12==12.6.77 in /usr/local/lib/python3.12/dist-packages (from tor
```

```
ch>=2.0.0->-r requirements.txt (line 8) (12.6.77)
Requirement already satisfied: nvidia-cuda-cupti-cu12==12.6.80 in /usr/local/lib/python3.12/dist-packages (from torch
>=2.0.0->-r requirements.txt (line 8)) (12.6.80)
Requirement already satisfied: nvidia-cudnn-cu12==9.10.2.21 in /usr/local/lib/python3.12/dist-packages (from torch>=
2.0.0->-r requirements.txt (line 8)) (9.10.2.21)
Requirement already satisfied: nvidia-cublas-cu12==12.6.4.1 in /usr/local/lib/python3.12/dist-packages (from torch>=
2.0.0->-r requirements.txt (line 8)) (12.6.4.1)
Requirement already satisfied: nvidia-cufft-cu12==11.3.0.4 in /usr/local/lib/python3.12/dist-packages (from torch>=2.
0.0->-r requirements.txt (line 8)) (11.3.0.4)
Requirement already satisfied: nvidia-curand-cu12==10.3.7.77 in /usr/local/lib/python3.12/dist-packages (from torch>=
2.0.0->-r requirements.txt (line 8)) (10.3.7.77)
Requirement already satisfied: nvidia-cusolver-cu12==11.7.1.2 in /usr/local/lib/python3.12/dist-packages (from torch>=
2.0.0->-r requirements.txt (line 8)) (11.7.1.2)
Requirement already satisfied: nvidia-cusparse-cu12==12.5.4.2 in /usr/local/lib/python3.12/dist-packages (from torch>=
2.0.0->-r requirements.txt (line 8)) (12.5.4.2)
Requirement already satisfied: nvidia-cusparseelt-cu12==0.7.1 in /usr/local/lib/python3.12/dist-packages (from torch>=
2.0.0->-r requirements.txt (line 8)) (0.7.1)
Requirement already satisfied: nvidia-nccl-cu12==2.27.5 in /usr/local/lib/python3.12/dist-packages (from torch>=2.0.0
->-r requirements.txt (line 8)) (2.27.5)
Requirement already satisfied: nvidia-nvshmem-cu12==3.3.20 in /usr/local/lib/python3.12/dist-packages (from torch>=2.
0.0->-r requirements.txt (line 8)) (3.3.20)
Requirement already satisfied: nvidia-nvtx-cu12==12.6.77 in /usr/local/lib/python3.12/dist-packages (from torch>=2.0.
0->-r requirements.txt (line 8)) (12.6.77)
Requirement already satisfied: nvidia-nvjitlink-cu12==12.6.85 in /usr/local/lib/python3.12/dist-packages (from torch>=
2.0.0->-r requirements.txt (line 8)) (12.6.85)
Requirement already satisfied: nvidia-cufile-cu12==1.11.1.6 in /usr/local/lib/python3.12/dist-packages (from torch>=
2.0.0->-r requirements.txt (line 8)) (1.11.1.6)
Requirement already satisfied: triton==3.5.0 in /usr/local/lib/python3.12/dist-packages (from torch>=2.0.0->-r requir
ements.txt (line 8)) (3.5.0)
Requirement already satisfied: huggingface-hub>=0.24 in /usr/local/lib/python3.12/dist-packages (from segmentation-mo
dels-pytorch>=0.3.3->-r requirements.txt (line 10)) (0.36.0)
Requirement already satisfied: safetensors>=0.3.1 in /usr/local/lib/python3.12/dist-packages (from segmentation-model
s-pytorch>=0.3.3->-r requirements.txt (line 10)) (0.7.0)
Requirement already satisfied: timm>=0.9 in /usr/local/lib/python3.12/dist-packages (from segmentation-models-pytorch
>=0.3.3->-r requirements.txt (line 10)) (1.0.22)
Requirement already satisfied: PyYAML in /usr/local/lib/python3.12/dist-packages (from albumentations>=1.3.1->-r requ
irements.txt (line 11)) (6.0.3)
Requirement already satisfied: pydantic>=2.9.2 in /usr/local/lib/python3.12/dist-packages (from albumentations>=1.3.1
->-r requirements.txt (line 11)) (2.12.3)
Requirement already satisfied: albucore==0.0.24 in /usr/local/lib/python3.12/dist-packages (from albumentations>=1.3.
1->-r requirements.txt (line 11)) (0.0.24)
Requirement already satisfied: opencv-python-headless>=4.9.0.80 in /usr/local/lib/python3.12/dist-packages (from albu
```

```
mentations>=1.3.1->-r requirements.txt (line 11)) (4.12.0.88)
Requirement already satisfied: stringzilla>=3.10.4 in /usr/local/lib/python3.12/dist-packages (from albucore==0.0.24->albumentations>=1.3.1->-r requirements.txt (line 11)) (4.2.3)
Requirement already satisfied: simsimd>=5.9.2 in /usr/local/lib/python3.12/dist-packages (from albucore==0.0.24->albumentations>=1.3.1->-r requirements.txt (line 11)) (6.5.3)
Requirement already satisfied: imageio!=2.35.0,>=2.33 in /usr/local/lib/python3.12/dist-packages (from scikit-image>=0.21.0->-r requirements.txt (line 15)) (2.37.2)
Requirement already satisfied: tifffile>=2022.8.12 in /usr/local/lib/python3.12/dist-packages (from scikit-image>=0.21.0->-r requirements.txt (line 15)) (2025.10.16)
Requirement already satisfied: packaging>=21 in /usr/local/lib/python3.12/dist-packages (from scikit-image>=0.21.0->-r requirements.txt (line 15)) (25.0)
Requirement already satisfied: lazy-loader>=0.4 in /usr/local/lib/python3.12/dist-packages (from scikit-image>=0.21.0->-r requirements.txt (line 15)) (0.4)
Requirement already satisfied: affine in /usr/local/lib/python3.12/dist-packages (from rasterio>=1.3.8->-r requirements.txt (line 19)) (2.4.0)
Requirement already satisfied: attrs in /usr/local/lib/python3.12/dist-packages (from rasterio>=1.3.8->-r requirements.txt (line 19)) (25.4.0)
Requirement already satisfied: certifi in /usr/local/lib/python3.12/dist-packages (from rasterio>=1.3.8->-r requirements.txt (line 19)) (2025.11.12)
Requirement already satisfied: click>=4.0 in /usr/local/lib/python3.12/dist-packages (from rasterio>=1.3.8->-r requirements.txt (line 19)) (8.3.1)
Requirement already satisfied: cligj>=0.5 in /usr/local/lib/python3.12/dist-packages (from rasterio>=1.3.8->-r requirements.txt (line 19)) (0.7.2)
Requirement already satisfied: click-plugins in /usr/local/lib/python3.12/dist-packages (from rasterio>=1.3.8->-r requirements.txt (line 19)) (1.1.1.2)
Requirement already satisfied: pyparsing in /usr/local/lib/python3.12/dist-packages (from rasterio>=1.3.8->-r requirements.txt (line 19)) (3.2.5)
Requirement already satisfied: filelock in /usr/local/lib/python3.12/dist-packages (from torch>=2.0.0->-r requirements.txt (line 8)) (3.20.0)
Requirement already satisfied: typing-extensions>=4.10.0 in /usr/local/lib/python3.12/dist-packages (from torch>=2.0.0->-r requirements.txt (line 8)) (4.15.0)
Requirement already satisfied: setuptools in /usr/local/lib/python3.12/dist-packages (from torch>=2.0.0->-r requirements.txt (line 8)) (75.2.0)
Requirement already satisfied: sympy>=1.13.3 in /usr/local/lib/python3.12/dist-packages (from torch>=2.0.0->-r requirements.txt (line 8)) (1.14.0)
Requirement already satisfied: networkx>=2.5.1 in /usr/local/lib/python3.12/dist-packages (from torch>=2.0.0->-r requirements.txt (line 8)) (3.6)
Requirement already satisfied: jinja2 in /usr/local/lib/python3.12/dist-packages (from torch>=2.0.0->-r requirements.txt (line 8)) (3.1.6)
Requirement already satisfied: fsspec>=0.8.5 in /usr/local/lib/python3.12/dist-packages (from torch>=2.0.0->-r requirements.txt (line 8)) (2025.3.0)
Requirement already satisfied: nvidia-cuda-nvrtc-cu12==12.6.77 in /usr/local/lib/python3.12/dist-packages (from torch
```

```
>=2.0.0->-r requirements.txt (line 8)) (12.6.77)
Requirement already satisfied: nvidia-cuda-runtime-cu12==12.6.77 in /usr/local/lib/python3.12/dist-packages (from torch>=2.0.0->-r requirements.txt (line 8)) (12.6.77)
Requirement already satisfied: nvidia-cuda-cupti-cu12==12.6.80 in /usr/local/lib/python3.12/dist-packages (from torch>=2.0.0->-r requirements.txt (line 8)) (12.6.80)
Requirement already satisfied: nvidia-cudnn-cu12==9.10.2.21 in /usr/local/lib/python3.12/dist-packages (from torch>=2.0.0->-r requirements.txt (line 8)) (9.10.2.21)
Requirement already satisfied: nvidia-cublas-cu12==12.6.4.1 in /usr/local/lib/python3.12/dist-packages (from torch>=2.0.0->-r requirements.txt (line 8)) (12.6.4.1)
Requirement already satisfied: nvidia-cufft-cu12==11.3.0.4 in /usr/local/lib/python3.12/dist-packages (from torch>=2.0.0->-r requirements.txt (line 8)) (11.3.0.4)
Requirement already satisfied: nvidia-curand-cu12==10.3.7.77 in /usr/local/lib/python3.12/dist-packages (from torch>=2.0.0->-r requirements.txt (line 8)) (10.3.7.77)
Requirement already satisfied: nvidia-cusolver-cu12==11.7.1.2 in /usr/local/lib/python3.12/dist-packages (from torch>=2.0.0->-r requirements.txt (line 8)) (11.7.1.2)
Requirement already satisfied: nvidia-cusparse-cu12==12.5.4.2 in /usr/local/lib/python3.12/dist-packages (from torch>=2.0.0->-r requirements.txt (line 8)) (12.5.4.2)
Requirement already satisfied: nvidia-cusparseelt-cu12==0.7.1 in /usr/local/lib/python3.12/dist-packages (from torch>=2.0.0->-r requirements.txt (line 8)) (0.7.1)
Requirement already satisfied: nvidia-nccl-cu12==2.27.5 in /usr/local/lib/python3.12/dist-packages (from torch>=2.0.0->-r requirements.txt (line 8)) (2.27.5)
Requirement already satisfied: nvidia-nvshmem-cu12==3.3.20 in /usr/local/lib/python3.12/dist-packages (from torch>=2.0.0->-r requirements.txt (line 8)) (3.3.20)
Requirement already satisfied: nvidia-nvtx-cu12==12.6.77 in /usr/local/lib/python3.12/dist-packages (from torch>=2.0.0->-r requirements.txt (line 8)) (12.6.77)
Requirement already satisfied: nvidia-nvjitlink-cu12==12.6.85 in /usr/local/lib/python3.12/dist-packages (from torch>=2.0.0->-r requirements.txt (line 8)) (12.6.85)
Requirement already satisfied: nvidia-cufile-cu12==1.11.1.6 in /usr/local/lib/python3.12/dist-packages (from torch>=2.0.0->-r requirements.txt (line 8)) (1.11.1.6)
Requirement already satisfied: triton==3.5.0 in /usr/local/lib/python3.12/dist-packages (from torch>=2.0.0->-r requirements.txt (line 8)) (3.5.0)
Requirement already satisfied: huggingface-hub>=0.24 in /usr/local/lib/python3.12/dist-packages (from segmentation-models-pytorch>=0.3.3->-r requirements.txt (line 10)) (0.36.0)
Requirement already satisfied: safetensors>=0.3.1 in /usr/local/lib/python3.12/dist-packages (from segmentation-models-pytorch>=0.3.3->-r requirements.txt (line 10)) (0.7.0)
Requirement already satisfied: timm>=0.9 in /usr/local/lib/python3.12/dist-packages (from segmentation-models-pytorch>=0.3.3->-r requirements.txt (line 10)) (1.0.22)
Requirement already satisfied: PyYAML in /usr/local/lib/python3.12/dist-packages (from albumentations>=1.3.1->-r requirements.txt (line 11)) (6.0.3)
Requirement already satisfied: pydantic>=2.9.2 in /usr/local/lib/python3.12/dist-packages (from albumentations>=1.3.1->-r requirements.txt (line 11)) (2.12.3)
Requirement already satisfied: albucore==0.0.24 in /usr/local/lib/python3.12/dist-packages (from albumentations>=1.3.
```

```
1->-r requirements.txt (line 11)) (0.0.24)
Requirement already satisfied: opencv-python-headless>=4.9.0.80 in /usr/local/lib/python3.12/dist-packages (from albumentations>=1.3.1->-r requirements.txt (line 11)) (4.12.0.88)
Requirement already satisfied: stringzilla>=3.10.4 in /usr/local/lib/python3.12/dist-packages (from albucore==0.0.24->albumentations>=1.3.1->-r requirements.txt (line 11)) (4.2.3)
Requirement already satisfied: simsimd>=5.9.2 in /usr/local/lib/python3.12/dist-packages (from albucore==0.0.24->albumentations>=1.3.1->-r requirements.txt (line 11)) (6.5.3)
Requirement already satisfied: imageio!=2.35.0,>=2.33 in /usr/local/lib/python3.12/dist-packages (from scikit-image>=0.21.0->-r requirements.txt (line 15)) (2.37.2)
Requirement already satisfied: tifffile>=2022.8.12 in /usr/local/lib/python3.12/dist-packages (from scikit-image>=0.21.0->-r requirements.txt (line 15)) (2025.10.16)
Requirement already satisfied: packaging>=21 in /usr/local/lib/python3.12/dist-packages (from scikit-image>=0.21.0->-r requirements.txt (line 15)) (25.0)
Requirement already satisfied: lazy-loader>=0.4 in /usr/local/lib/python3.12/dist-packages (from scikit-image>=0.21.0->-r requirements.txt (line 15)) (0.4)
Requirement already satisfied: affine in /usr/local/lib/python3.12/dist-packages (from rasterio>=1.3.8->-r requirements.txt (line 19)) (2.4.0)
Requirement already satisfied: attrs in /usr/local/lib/python3.12/dist-packages (from rasterio>=1.3.8->-r requirements.txt (line 19)) (25.4.0)
Requirement already satisfied: certifi in /usr/local/lib/python3.12/dist-packages (from rasterio>=1.3.8->-r requirements.txt (line 19)) (2025.11.12)
Requirement already satisfied: click>=4.0 in /usr/local/lib/python3.12/dist-packages (from rasterio>=1.3.8->-r requirements.txt (line 19)) (8.3.1)
Requirement already satisfied: cligj>=0.5 in /usr/local/lib/python3.12/dist-packages (from rasterio>=1.3.8->-r requirements.txt (line 19)) (0.7.2)
Requirement already satisfied: click-plugins in /usr/local/lib/python3.12/dist-packages (from rasterio>=1.3.8->-r requirements.txt (line 19)) (1.1.1.2)
Requirement already satisfied: pyparsing in /usr/local/lib/python3.12/dist-packages (from rasterio>=1.3.8->-r requirements.txt (line 19)) (3.2.5)
Requirement already satisfied: pyogrio>=0.7.2 in /usr/local/lib/python3.12/dist-packages (from geopandas>=0.13.0->-r requirements.txt (line 20)) (0.11.1)
Requirement already satisfied: pyproj>=3.5.0 in /usr/local/lib/python3.12/dist-packages (from geopandas>=0.13.0->-r requirements.txt (line 20)) (3.7.2)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.12/dist-packages (from matplotlib>=3.7.0->-r requirements.txt (line 24)) (1.3.3)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.12/dist-packages (from matplotlib>=3.7.0->-r requirements.txt (line 24)) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.12/dist-packages (from matplotlib>=3.7.0->-r requirements.txt (line 24)) (4.60.1)
Requirement already satisfied: kiwisolver>=1.3.1 in /usr/local/lib/python3.12/dist-packages (from matplotlib>=3.7.0->-r requirements.txt (line 24)) (1.4.9)
Requirement already satisfied: absl-py>=0.4 in /usr/local/lib/python3.12/dist-packages (from tensorboard>=2.13.0->-r
```

```
requirements.txt (line 31)) (1.4.0)
Requirement already satisfied: grpcio>=1.48.2 in /usr/local/lib/python3.12/dist-packages (from tensorboard>=2.13.0--r requirements.txt (line 31)) (1.76.0)
Requirement already satisfied: markdown>=2.6.8 in /usr/local/lib/python3.12/dist-packages (from tensorboard>=2.13.0--r requirements.txt (line 31)) (3.10)
Requirement already satisfied: protobuf!=4.24.0,>=3.19.6 in /usr/local/lib/python3.12/dist-packages (from tensorboard>=2.13.0--r requirements.txt (line 31)) (5.29.5)
Requirement already satisfied: six>1.9 in /usr/local/lib/python3.12/dist-packages (from tensorboard>=2.13.0--r requirements.txt (line 31)) (1.17.0)
Requirement already satisfied: tensorboard-data-server<0.8.0,>=0.7.0 in /usr/local/lib/python3.12/dist-packages (from tensorboard>=2.13.0--r requirements.txt (line 31)) (0.7.2)
Requirement already satisfied: werkzeug>=1.0.1 in /usr/local/lib/python3.12/dist-packages (from tensorboard>=2.13.0--r requirements.txt (line 31)) (3.1.3)
Requirement already satisfied: notebook in /usr/local/lib/python3.12/dist-packages (from jupyter>=1.0.0--r requirements.txt (line 34)) (7.5.0)
Requirement already satisfied: jupyter-console in /usr/local/lib/python3.12/dist-packages (from jupyter>=1.0.0--r requirements.txt (line 34)) (6.6.3)
Requirement already satisfied: nbconvert in /usr/local/lib/python3.12/dist-packages (from jupyter>=1.0.0--r requirements.txt (line 34)) (7.16.6)
Requirement already satisfied: jupyterlab in /usr/local/lib/python3.12/dist-packages (from jupyter>=1.0.0--r requirements.txt (line 34)) (4.5.0)
Requirement already satisfied: comm>=0.1.1 in /usr/local/lib/python3.12/dist-packages (from ipykernel>=6.25.0--r requirements.txt (line 35)) (0.2.3)
Requirement already satisfied: debugpy>=1.6.5 in /usr/local/lib/python3.12/dist-packages (from ipykernel>=6.25.0--r requirements.txt (line 35)) (1.8.15)
Requirement already satisfied: ipython>=7.23.1 in /usr/local/lib/python3.12/dist-packages (from ipykernel>=6.25.0--r requirements.txt (line 35)) (7.34.0)
Requirement already satisfied: jupyter-client>=8.0.0 in /usr/local/lib/python3.12/dist-packages (from ipykernel>=6.25.0--r requirements.txt (line 35)) (8.6.3)
Requirement already satisfied: jupyter-core!=5.0.*,>=4.12 in /usr/local/lib/python3.12/dist-packages (from ipykernel>=6.25.0--r requirements.txt (line 35)) (5.9.1)
Requirement already satisfied: matplotlib-inline>=0.1 in /usr/local/lib/python3.12/dist-packages (from ipykernel>=6.25.0--r requirements.txt (line 35)) (0.2.1)
Requirement already satisfied: nest-asyncio>=1.4 in /usr/local/lib/python3.12/dist-packages (from ipykernel>=6.25.0--r requirements.txt (line 35)) (1.6.0)
Requirement already satisfied: psutil>=5.7 in /usr/local/lib/python3.12/dist-packages (from ipykernel>=6.25.0--r requirements.txt (line 35)) (5.9.5)
Requirement already satisfied: pyzmq>=25 in /usr/local/lib/python3.12/dist-packages (from ipykernel>=6.25.0--r requirements.txt (line 35)) (26.2.1)
Requirement already satisfied: tornado>=6.2 in /usr/local/lib/python3.12/dist-packages (from ipykernel>=6.25.0--r requirements.txt (line 35)) (6.5.1)
Requirement already satisfied: traitlets>=5.4.0 in /usr/local/lib/python3.12/dist-packages (from ipykernel>=6.25.0--r
```

```
r requirements.txt (line 35)) (5.7.1)
Requirement already satisfied: widgetsnbextension~=4.0.14 in /usr/local/lib/python3.12/dist-packages (from ipywidgets
>=8.1.0->-r requirements.txt (line 36)) (4.0.15)
Requirement already satisfied: jupyterlab_widgets~=3.0.15 in /usr/local/lib/python3.12/dist-packages (from ipywidgets
>=8.1.0->-r requirements.txt (line 36)) (3.0.16)
Requirement already satisfied: requests in /usr/local/lib/python3.12/dist-packages (from huggingface-hub>=0.24->segme
ntation-models-pytorch>=0.3.3->-r requirements.txt (line 10)) (2.32.4)
Requirement already satisfied: hf-xet<2.0.0,>=1.1.3 in /usr/local/lib/python3.12/dist-packages (from huggingface-hub>
=0.24->segmentation-models-pytorch>=0.3.3->-r requirements.txt (line 10)) (1.2.0)
Requirement already satisfied: jedi>=0.16 in /usr/local/lib/python3.12/dist-packages (from ipython>=7.23.1->ipykernel
>=6.25.0->-r requirements.txt (line 35)) (0.19.2)
Requirement already satisfied: decorator in /usr/local/lib/python3.12/dist-packages (from ipython>=7.23.1->ipykernel>
=6.25.0->-r requirements.txt (line 35)) (4.4.2)
Requirement already satisfied: pickleshare in /usr/local/lib/python3.12/dist-packages (from ipython>=7.23.1->ipykerne
l>=6.25.0->-r requirements.txt (line 35)) (0.7.5)
Requirement already satisfied: prompt-toolkit!=3.0.0,!>=3.0.1,<3.1.0,>=2.0.0 in /usr/local/lib/python3.12/dist-package
s (from ipython>=7.23.1->ipykernel>=6.25.0->-r requirements.txt (line 35)) (3.0.52)
Requirement already satisfied: pygments in /usr/local/lib/python3.12/dist-packages (from ipython>=7.23.1->ipykernel>=
6.25.0->-r requirements.txt (line 35)) (2.19.2)
Requirement already satisfied: backcall in /usr/local/lib/python3.12/dist-packages (from ipython>=7.23.1->ipykernel>=
6.25.0->-r requirements.txt (line 35)) (0.2.0)
Requirement already satisfied: pexpect>4.3 in /usr/local/lib/python3.12/dist-packages (from ipython>=7.23.1->ipykerne
l>=6.25.0->-r requirements.txt (line 35)) (4.9.0)
Requirement already satisfied: pyogrio>=0.7.2 in /usr/local/lib/python3.12/dist-packages (from geopandas>=0.13.0->-r
requirements.txt (line 20)) (0.11.1)
Requirement already satisfied: pyproj>=3.5.0 in /usr/local/lib/python3.12/dist-packages (from geopandas>=0.13.0->-r r
equirements.txt (line 20)) (3.7.2)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.12/dist-packages (from matplotlib>=3.7.0->-
r requirements.txt (line 24)) (1.3.3)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.12/dist-packages (from matplotlib>=3.7.0->-r re
quirements.txt (line 24)) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.12/dist-packages (from matplotlib>=3.7.0->-
r requirements.txt (line 24)) (4.60.1)
Requirement already satisfied: kiwisolver>=1.3.1 in /usr/local/lib/python3.12/dist-packages (from matplotlib>=3.7.0->-
r requirements.txt (line 24)) (1.4.9)
Requirement already satisfied: absl-py>=0.4 in /usr/local/lib/python3.12/dist-packages (from tensorboard>=2.13.0->-r
requirements.txt (line 31)) (1.4.0)
Requirement already satisfied: grpcio>=1.48.2 in /usr/local/lib/python3.12/dist-packages (from tensorboard>=2.13.0->-
r requirements.txt (line 31)) (1.76.0)
Requirement already satisfied: markdown>=2.6.8 in /usr/local/lib/python3.12/dist-packages (from tensorboard>=2.13.0->-
r requirements.txt (line 31)) (3.10)
Requirement already satisfied: protobuf!=4.24.0,>=3.19.6 in /usr/local/lib/python3.12/dist-packages (from tensorboard
```

```
>=2.13.0->-r requirements.txt (line 31)) (5.29.5)
Requirement already satisfied: six>1.9 in /usr/local/lib/python3.12/dist-packages (from tensorboard>=2.13.0->-r requirements.txt (line 31)) (1.17.0)
Requirement already satisfied: tensorflow-data-server<0.8.0,>=0.7.0 in /usr/local/lib/python3.12/dist-packages (from tensorboard>=2.13.0->-r requirements.txt (line 31)) (0.7.2)
Requirement already satisfied: werkzeug>=1.0.1 in /usr/local/lib/python3.12/dist-packages (from tensorboard>=2.13.0->-r requirements.txt (line 31)) (3.1.3)
Requirement already satisfied: notebook in /usr/local/lib/python3.12/dist-packages (from jupyter>=1.0.0->-r requirements.txt (line 34)) (7.5.0)
Requirement already satisfied: jupyter-console in /usr/local/lib/python3.12/dist-packages (from jupyter>=1.0.0->-r requirements.txt (line 34)) (6.6.3)
Requirement already satisfied: nbconvert in /usr/local/lib/python3.12/dist-packages (from jupyter>=1.0.0->-r requirements.txt (line 34)) (7.16.6)
Requirement already satisfied: jupyterlab in /usr/local/lib/python3.12/dist-packages (from jupyter>=1.0.0->-r requirements.txt (line 34)) (4.5.0)
Requirement already satisfied: comm>=0.1.1 in /usr/local/lib/python3.12/dist-packages (from ipykernel>=6.25.0->-r requirements.txt (line 35)) (0.2.3)
Requirement already satisfied: debugpy>=1.6.5 in /usr/local/lib/python3.12/dist-packages (from ipykernel>=6.25.0->-r requirements.txt (line 35)) (1.8.15)
Requirement already satisfied: ipython>=7.23.1 in /usr/local/lib/python3.12/dist-packages (from ipykernel>=6.25.0->-r requirements.txt (line 35)) (7.34.0)
Requirement already satisfied: jupyter-client>=8.0.0 in /usr/local/lib/python3.12/dist-packages (from ipykernel>=6.25.0->-r requirements.txt (line 35)) (8.6.3)
Requirement already satisfied: jupyter-core!=5.0.*,>=4.12 in /usr/local/lib/python3.12/dist-packages (from ipykernel>=6.25.0->-r requirements.txt (line 35)) (5.9.1)
Requirement already satisfied: matplotlib-inline>=0.1 in /usr/local/lib/python3.12/dist-packages (from ipykernel>=6.25.0->-r requirements.txt (line 35)) (0.2.1)
Requirement already satisfied: nest-asyncio>=1.4 in /usr/local/lib/python3.12/dist-packages (from ipykernel>=6.25.0->-r requirements.txt (line 35)) (1.6.0)
Requirement already satisfied: psutil>=5.7 in /usr/local/lib/python3.12/dist-packages (from ipykernel>=6.25.0->-r requirements.txt (line 35)) (5.9.5)
Requirement already satisfied: pyzmq>=25 in /usr/local/lib/python3.12/dist-packages (from ipykernel>=6.25.0->-r requirements.txt (line 35)) (26.2.1)
Requirement already satisfied: tornado>=6.2 in /usr/local/lib/python3.12/dist-packages (from ipykernel>=6.25.0->-r requirements.txt (line 35)) (6.5.1)
Requirement already satisfied: traitlets>=5.4.0 in /usr/local/lib/python3.12/dist-packages (from ipykernel>=6.25.0->-r requirements.txt (line 35)) (5.7.1)
Requirement already satisfied: widgetsnbextension~=4.0.14 in /usr/local/lib/python3.12/dist-packages (from ipywidgets>=8.1.0->-r requirements.txt (line 36)) (4.0.15)
Requirement already satisfied: jupyterlab_widgets~=3.0.15 in /usr/local/lib/python3.12/dist-packages (from ipywidgets>=8.1.0->-r requirements.txt (line 36)) (3.0.16)
Requirement already satisfied: requests in /usr/local/lib/python3.12/dist-packages (from huggingface-hub>=0.24->segme
```

```
ntation-models-pytorch>=0.3.3->-r requirements.txt (line 10)) (2.32.4)
Requirement already satisfied: hf-xet<2.0.0,>=1.1.3 in /usr/local/lib/python3.12/dist-packages (from huggingface-hub>=0.24->segmentation-models-pytorch>=0.3.3->-r requirements.txt (line 10)) (1.2.0)
Requirement already satisfied: jedi>=0.16 in /usr/local/lib/python3.12/dist-packages (from ipython>=7.23.1->ipykernel>=6.25.0->-r requirements.txt (line 35)) (0.19.2)
Requirement already satisfied: decorator in /usr/local/lib/python3.12/dist-packages (from ipython>=7.23.1->ipykernel>=6.25.0->-r requirements.txt (line 35)) (4.4.2)
Requirement already satisfied: pickleshare in /usr/local/lib/python3.12/dist-packages (from ipython>=7.23.1->ipykerne1>=6.25.0->-r requirements.txt (line 35)) (0.7.5)
Requirement already satisfied: prompt-toolkit!=3.0.0,!>=3.0.1,<3.1.0,>=2.0.0 in /usr/local/lib/python3.12/dist-packages (from ipython>=7.23.1->ipykernel>=6.25.0->-r requirements.txt (line 35)) (3.0.52)
Requirement already satisfied: pygments in /usr/local/lib/python3.12/dist-packages (from ipython>=7.23.1->ipykernel>=6.25.0->-r requirements.txt (line 35)) (2.19.2)
Requirement already satisfied: backcall in /usr/local/lib/python3.12/dist-packages (from ipython>=7.23.1->ipykernel>=6.25.0->-r requirements.txt (line 35)) (0.2.0)
Requirement already satisfied: pexpect>4.3 in /usr/local/lib/python3.12/dist-packages (from ipython>=7.23.1->ipykerne1>=6.25.0->-r requirements.txt (line 35)) (4.9.0)
Requirement already satisfied: platformdirs>=2.5 in /usr/local/lib/python3.12/dist-packages (from jupyter-core!=5.0.*,>=4.12->ipykernel>=6.25.0->-r requirements.txt (line 35)) (4.5.0)
Requirement already satisfied: annotated-types>=0.6.0 in /usr/local/lib/python3.12/dist-packages (from pydantic>=2.9.2->albumentations>=1.3.1->-r requirements.txt (line 11)) (0.7.0)
Requirement already satisfied: pydantic-core==2.41.4 in /usr/local/lib/python3.12/dist-packages (from pydantic>=2.9.2->albumentations>=1.3.1->-r requirements.txt (line 11)) (2.41.4)
Requirement already satisfied: typing-inspection>=0.4.2 in /usr/local/lib/python3.12/dist-packages (from pydantic>=2.9.2->albumentations>=1.3.1->-r requirements.txt (line 11)) (0.4.2)
Requirement already satisfied: mpmath<1.4,>=1.1.0 in /usr/local/lib/python3.12/dist-packages (from sympy>=1.13.3->torch>=2.0.0->-r requirements.txt (line 8)) (1.3.0)
Requirement already satisfied: MarkupSafe>=2.1.1 in /usr/local/lib/python3.12/dist-packages (from werkzeug>=1.0.1->tensordboard>=2.13.0->-r requirements.txt (line 31)) (3.0.3)
Requirement already satisfied: async-lru>=1.0.0 in /usr/local/lib/python3.12/dist-packages (from jupyterlab->jupyter>=1.0.0->-r requirements.txt (line 34)) (2.0.5)
Requirement already satisfied: httpx<1,>=0.25.0 in /usr/local/lib/python3.12/dist-packages (from jupyterlab->jupyter>=1.0.0->-r requirements.txt (line 34)) (0.28.1)
Requirement already satisfied: jupyter-lsp>=2.0.0 in /usr/local/lib/python3.12/dist-packages (from jupyterlab->jupyter>=1.0.0->-r requirements.txt (line 34)) (2.3.0)
Requirement already satisfied: jupyter-server<3,>=2.4.0 in /usr/local/lib/python3.12/dist-packages (from jupyterlab->jupyter>=1.0.0->-r requirements.txt (line 34)) (2.14.0)
Requirement already satisfied: jupyterlab-server<3,>=2.28.0 in /usr/local/lib/python3.12/dist-packages (from jupyterlab->jupyter>=1.0.0->-r requirements.txt (line 34)) (2.28.0)
Requirement already satisfied: notebook-shim>=0.2 in /usr/local/lib/python3.12/dist-packages (from jupyterlab->jupyter>=1.0.0->-r requirements.txt (line 34)) (0.2.4)
Requirement already satisfied: beautifulsoup4 in /usr/local/lib/python3.12/dist-packages (from nbconvert->jupyter>=1.
```

```
0.0->-r requirements.txt (line 34)) (4.13.5)
Requirement already satisfied: bleach!=5.0.0 in /usr/local/lib/python3.12/dist-packages (from bleach[css]!=5.0.0->nbc
onvert->jupyter>=1.0.0->-r requirements.txt (line 34)) (6.3.0)
Requirement already satisfied: defusedxml in /usr/local/lib/python3.12/dist-packages (from nbconvert->jupyter>=1.0.0-
->-r requirements.txt (line 34)) (0.7.1)
Requirement already satisfied: jupyterlab-pygments in /usr/local/lib/python3.12/dist-packages (from nbconvert->jupyter
r>=1.0.0->-r requirements.txt (line 34)) (0.3.0)
Requirement already satisfied: mistune<4,>=2.0.3 in /usr/local/lib/python3.12/dist-packages (from nbconvert->jupyter>
=1.0.0->-r requirements.txt (line 34)) (3.1.4)
Requirement already satisfied: nbclient>=0.5.0 in /usr/local/lib/python3.12/dist-packages (from nbconvert->jupyter>=
1.0.0->-r requirements.txt (line 34)) (0.10.2)
Requirement already satisfied: nbformat>=5.7 in /usr/local/lib/python3.12/dist-packages (from nbconvert->jupyter>=1.
0.0->-r requirements.txt (line 34)) (5.10.4)
Requirement already satisfied: pandocfilters>=1.4.1 in /usr/local/lib/python3.12/dist-packages (from nbconvert->jupy
ter>=1.0.0->-r requirements.txt (line 34)) (1.5.1)
Requirement already satisfied: webencodings in /usr/local/lib/python3.12/dist-packages (from bleach!=5.0.0->bleach[cs
s]!=5.0.0->nbconvert->jupyter>=1.0.0->-r requirements.txt (line 34)) (0.5.1)
Requirement already satisfied: tinycc2<1.5,>=1.1.0 in /usr/local/lib/python3.12/dist-packages (from bleach[css]!=5.
0.0->nbconvert->jupyter>=1.0.0->-r requirements.txt (line 34)) (1.4.0)
Requirement already satisfied: anyio in /usr/local/lib/python3.12/dist-packages (from httpx<1,>=0.25.0->jupyterlab->j
upyter>=1.0.0->-r requirements.txt (line 34)) (4.11.0)
Requirement already satisfied: httpcore==1.* in /usr/local/lib/python3.12/dist-packages (from httpx<1,>=0.25.0->jupy
terlab->jupyter>=1.0.0->-r requirements.txt (line 34)) (1.0.9)
Requirement already satisfied: idna in /usr/local/lib/python3.12/dist-packages (from httpx<1,>=0.25.0->jupyterlab->ju
pyter>=1.0.0->-r requirements.txt (line 34)) (3.11)
Requirement already satisfied: platformdirs>=2.5 in /usr/local/lib/python3.12/dist-packages (from jupyter-core!=5.0.
*,>=4.12->ipykernel>=6.25.0->-r requirements.txt (line 35)) (4.5.0)
Requirement already satisfied: annotated-types>=0.6.0 in /usr/local/lib/python3.12/dist-packages (from pydantic>=2.9.
2->albumentations>=1.3.1->-r requirements.txt (line 11)) (0.7.0)
Requirement already satisfied: pydantic-core==2.41.4 in /usr/local/lib/python3.12/dist-packages (from pydantic>=2.9.2
->albumentations>=1.3.1->-r requirements.txt (line 11)) (2.41.4)
Requirement already satisfied: typing-inspection>=0.4.2 in /usr/local/lib/python3.12/dist-packages (from pydantic>=2.
9.2->albumentations>=1.3.1->-r requirements.txt (line 11)) (0.4.2)
Requirement already satisfied: mpmath<1.4,>=1.1.0 in /usr/local/lib/python3.12/dist-packages (from sympy>=1.13.3->tor
ch>=2.0.0->-r requirements.txt (line 8)) (1.3.0)
Requirement already satisfied: MarkupSafe>=2.1.1 in /usr/local/lib/python3.12/dist-packages (from werkzeug>=1.0.1->te
nsorboard>=2.13.0->-r requirements.txt (line 31)) (3.0.3)
Requirement already satisfied: async-lru>=1.0.0 in /usr/local/lib/python3.12/dist-packages (from jupyterlab->jupyter>
=1.0.0->-r requirements.txt (line 34)) (2.0.5)
Requirement already satisfied: httpx<1,>=0.25.0 in /usr/local/lib/python3.12/dist-packages (from jupyterlab->jupyter>
=1.0.0->-r requirements.txt (line 34)) (0.28.1)
Requirement already satisfied: jupyter-lsp>=2.0.0 in /usr/local/lib/python3.12/dist-packages (from jupyterlab->jupyte
```

```
r>=1.0.0->-r requirements.txt (line 34)) (2.3.0)
Requirement already satisfied: jupyter-server<3,>=2.4.0 in /usr/local/lib/python3.12/dist-packages (from jupyterlab->jupyter>=1.0.0->-r requirements.txt (line 34)) (2.14.0)
Requirement already satisfied: jupyterlab-server<3,>=2.28.0 in /usr/local/lib/python3.12/dist-packages (from jupyterlab->jupyter>=1.0.0->-r requirements.txt (line 34)) (2.28.0)
Requirement already satisfied: notebook-shim>=0.2 in /usr/local/lib/python3.12/dist-packages (from jupyterlab->jupyter>=1.0.0->-r requirements.txt (line 34)) (0.2.4)
Requirement already satisfied: beautifulsoup4 in /usr/local/lib/python3.12/dist-packages (from nbconvert->jupyter>=1.0.0->-r requirements.txt (line 34)) (4.13.5)
Requirement already satisfied: bleach!=5.0.0 in /usr/local/lib/python3.12/dist-packages (from bleach[css]!=5.0.0->nbconvert->jupyter>=1.0.0->-r requirements.txt (line 34)) (6.3.0)
Requirement already satisfied: defusedxml in /usr/local/lib/python3.12/dist-packages (from nbconvert->jupyter>=1.0.0->-r requirements.txt (line 34)) (0.7.1)
Requirement already satisfied: jupyterlab-pygments in /usr/local/lib/python3.12/dist-packages (from nbconvert->jupyter>=1.0.0->-r requirements.txt (line 34)) (0.3.0)
Requirement already satisfied: mistune<4,>=2.0.3 in /usr/local/lib/python3.12/dist-packages (from nbconvert->jupyter>=1.0.0->-r requirements.txt (line 34)) (3.1.4)
Requirement already satisfied: nbclient>=0.5.0 in /usr/local/lib/python3.12/dist-packages (from nbconvert->jupyter>=1.0.0->-r requirements.txt (line 34)) (0.10.2)
Requirement already satisfied: nbformat>=5.7 in /usr/local/lib/python3.12/dist-packages (from nbconvert->jupyter>=1.0.0->-r requirements.txt (line 34)) (5.10.4)
Requirement already satisfied: pandocfilters>=1.4.1 in /usr/local/lib/python3.12/dist-packages (from nbconvert->jupyter>=1.0.0->-r requirements.txt (line 34)) (1.5.1)
Requirement already satisfied: webencodings in /usr/local/lib/python3.12/dist-packages (from bleach!=5.0.0->bleach[css]!=5.0.0->nbconvert->jupyter>=1.0.0->-r requirements.txt (line 34)) (0.5.1)
Requirement already satisfied: tinycc2<1.5,>=1.1.0 in /usr/local/lib/python3.12/dist-packages (from bleach[css]!=5.0.0->nbconvert->jupyter>=1.0.0->-r requirements.txt (line 34)) (1.4.0)
Requirement already satisfied: anyio in /usr/local/lib/python3.12/dist-packages (from httpx<1,>=0.25.0->jupyterlab->jupyter>=1.0.0->-r requirements.txt (line 34)) (4.11.0)
Requirement already satisfied: httpcore==1.* in /usr/local/lib/python3.12/dist-packages (from httpx<1,>=0.25.0->jupyterlab->jupyter>=1.0.0->-r requirements.txt (line 34)) (1.0.9)
Requirement already satisfied: idna in /usr/local/lib/python3.12/dist-packages (from httpx<1,>=0.25.0->jupyterlab->jupyter>=1.0.0->-r requirements.txt (line 34)) (3.11)
Requirement already satisfied: h11>=0.16 in /usr/local/lib/python3.12/dist-packages (from httpcore==1.*->httpx<1,>=0.25.0->jupyterlab->jupyter>=1.0.0->-r requirements.txt (line 34)) (0.16.0)
Requirement already satisfied: parso<0.9.0,>=0.8.4 in /usr/local/lib/python3.12/dist-packages (from jedi>=0.16->ipython>=7.23.1->ipykernel>=6.25.0->-r requirements.txt (line 35)) (0.8.5)
Requirement already satisfied: argon2-cffi>=21.1 in /usr/local/lib/python3.12/dist-packages (from jupyter-server<3,>=2.4.0->jupyterlab->jupyter>=1.0.0->-r requirements.txt (line 34)) (25.1.0)
Requirement already satisfied: jupyter-events>=0.9.0 in /usr/local/lib/python3.12/dist-packages (from jupyter-server<3,>=2.4.0->jupyterlab->jupyter>=1.0.0->-r requirements.txt (line 34)) (0.12.0)
Requirement already satisfied: jupyter-server-terminals>=0.4.4 in /usr/local/lib/python3.12/dist-packages (from jupyter
```

```
er-server<3,>=2.4.0->jupyterlab->jupyter>=1.0.0->-r requirements.txt (line 34)) (0.5.3)
Requirement already satisfied: overrides>=5.0 in /usr/local/lib/python3.12/dist-packages (from jupyter-server<3,>=2.4.0->jupyterlab->jupyter>=1.0.0->-r requirements.txt (line 34)) (7.7.0)
Requirement already satisfied: prometheus-client>=0.9 in /usr/local/lib/python3.12/dist-packages (from jupyter-server<3,>=2.4.0->jupyterlab->jupyter>=1.0.0->-r requirements.txt (line 34)) (0.23.1)
Requirement already satisfied: send2trash>=1.8.2 in /usr/local/lib/python3.12/dist-packages (from jupyter-server<3,>=2.4.0->jupyterlab->jupyter>=1.0.0->-r requirements.txt (line 34)) (1.8.3)
Requirement already satisfied: terminado>=0.8.3 in /usr/local/lib/python3.12/dist-packages (from jupyter-server<3,>=2.4.0->jupyterlab->jupyter>=1.0.0->-r requirements.txt (line 34)) (0.18.1)
Requirement already satisfied: websocket-client>=1.7 in /usr/local/lib/python3.12/dist-packages (from jupyter-server<3,>=2.4.0->jupyterlab->jupyter>=1.0.0->-r requirements.txt (line 34)) (1.9.0)
Requirement already satisfied: babel>=2.10 in /usr/local/lib/python3.12/dist-packages (from jupyterlab-server<3,>=2.28.0->jupyterlab->jupyter>=1.0.0->-r requirements.txt (line 34)) (2.17.0)
Requirement already satisfied: json5>=0.9.0 in /usr/local/lib/python3.12/dist-packages (from jupyterlab-server<3,>=2.28.0->jupyterlab->jupyter>=1.0.0->-r requirements.txt (line 34)) (0.12.1)
Requirement already satisfied: jsonschema>=4.18.0 in /usr/local/lib/python3.12/dist-packages (from jupyterlab-server<3,>=2.28.0->jupyterlab->jupyter>=1.0.0->-r requirements.txt (line 34)) (4.25.1)
Requirement already satisfied: fastjsonschema>=2.15 in /usr/local/lib/python3.12/dist-packages (from nbformat>=5.7->nbconvert->jupyter>=1.0.0->-r requirements.txt (line 34)) (2.21.2)
Requirement already satisfied: ptyprocess>=0.5 in /usr/local/lib/python3.12/dist-packages (from pexpect>4.3->ipython>=7.23.1->ipykernel>=6.25.0->-r requirements.txt (line 35)) (0.7.0)
Requirement already satisfied: wcwidth in /usr/local/lib/python3.12/dist-packages (from prompt-toolkit!=3.0.0,!=3.0.1,<3.1.0,>=2.0.0->ipython>=7.23.1->ipykernel>=6.25.0->-r requirements.txt (line 35)) (0.2.14)
Requirement already satisfied: charset_normalizer<4,>=2 in /usr/local/lib/python3.12/dist-packages (from requests->huggingface-hub>=0.24->segmentation-models-pytorch>=0.3.3->-r requirements.txt (line 10)) (3.4.4)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.12/dist-packages (from requests->huggingface-hub>=0.24->segmentation-models-pytorch>=0.3.3->-r requirements.txt (line 10)) (2.5.0)
Requirement already satisfied: soupsieve>1.2 in /usr/local/lib/python3.12/dist-packages (from beautifulsoup4->nbconvert->jupyter>=1.0.0->-r requirements.txt (line 34)) (2.8)
Requirement already satisfied: sniffio>=1.1 in /usr/local/lib/python3.12/dist-packages (from anyio->httpx<1,>=0.25.0->jupyterlab->jupyter>=1.0.0->-r requirements.txt (line 34)) (1.3.1)
Requirement already satisfied: argon2-cffi-bindings in /usr/local/lib/python3.12/dist-packages (from argon2-cffi>=21.1->jupyter-server<3,>=2.4.0->jupyterlab->jupyter>=1.0.0->-r requirements.txt (line 34)) (25.1.0)
Requirement already satisfied: h11>=0.16 in /usr/local/lib/python3.12/dist-packages (from httpcore==1.*->httpx<1,>=0.25.0->jupyterlab->jupyter>=1.0.0->-r requirements.txt (line 34)) (0.16.0)
Requirement already satisfied: parso<0.9.0,>=0.8.4 in /usr/local/lib/python3.12/dist-packages (from jedi>=0.16->ipython>=7.23.1->ipykernel>=6.25.0->-r requirements.txt (line 35)) (0.8.5)
Requirement already satisfied: argon2-cffi>=21.1 in /usr/local/lib/python3.12/dist-packages (from jupyter-server<3,>=2.4.0->jupyterlab->jupyter>=1.0.0->-r requirements.txt (line 34)) (25.1.0)
Requirement already satisfied: jupyter-events>=0.9.0 in /usr/local/lib/python3.12/dist-packages (from jupyter-server<3,>=2.4.0->jupyterlab->jupyter>=1.0.0->-r requirements.txt (line 34)) (0.12.0)
Requirement already satisfied: jupyter-server-terminals>=0.4.4 in /usr/local/lib/python3.12/dist-packages (from jupyter-server-terminals>=0.4.4->jupyterlab->jupyter>=1.0.0->-r requirements.txt (line 34)) (0.4.4)
```

```
er-server<3,>=2.4.0->jupyterlab->jupyter>=1.0.0->-r requirements.txt (line 34)) (0.5.3)
Requirement already satisfied: overrides>=5.0 in /usr/local/lib/python3.12/dist-packages (from jupyter-server<3,>=2.4.0->jupyterlab->jupyter>=1.0.0->-r requirements.txt (line 34)) (7.7.0)
Requirement already satisfied: prometheus-client>=0.9 in /usr/local/lib/python3.12/dist-packages (from jupyter-server<3,>=2.4.0->jupyterlab->jupyter>=1.0.0->-r requirements.txt (line 34)) (0.23.1)
Requirement already satisfied: send2trash>=1.8.2 in /usr/local/lib/python3.12/dist-packages (from jupyter-server<3,>=2.4.0->jupyterlab->jupyter>=1.0.0->-r requirements.txt (line 34)) (1.8.3)
Requirement already satisfied: terminado>=0.8.3 in /usr/local/lib/python3.12/dist-packages (from jupyter-server<3,>=2.4.0->jupyterlab->jupyter>=1.0.0->-r requirements.txt (line 34)) (0.18.1)
Requirement already satisfied: websocket-client>=1.7 in /usr/local/lib/python3.12/dist-packages (from jupyter-server<3,>=2.4.0->jupyterlab->jupyter>=1.0.0->-r requirements.txt (line 34)) (1.9.0)
Requirement already satisfied: babel>=2.10 in /usr/local/lib/python3.12/dist-packages (from jupyterlab-server<3,>=2.28.0->jupyterlab->jupyter>=1.0.0->-r requirements.txt (line 34)) (2.17.0)
Requirement already satisfied: json5>=0.9.0 in /usr/local/lib/python3.12/dist-packages (from jupyterlab-server<3,>=2.28.0->jupyterlab->jupyter>=1.0.0->-r requirements.txt (line 34)) (0.12.1)
Requirement already satisfied: jsonschema>=4.18.0 in /usr/local/lib/python3.12/dist-packages (from jupyterlab-server<3,>=2.28.0->jupyterlab->jupyter>=1.0.0->-r requirements.txt (line 34)) (4.25.1)
Requirement already satisfied: fastjsonschema>=2.15 in /usr/local/lib/python3.12/dist-packages (from nbformat>=5.7->nbconvert->jupyter>=1.0.0->-r requirements.txt (line 34)) (2.21.2)
Requirement already satisfied: ptyprocess>=0.5 in /usr/local/lib/python3.12/dist-packages (from pexpect>4.3->ipython>=7.23.1->ipykernel>=6.25.0->-r requirements.txt (line 35)) (0.7.0)
Requirement already satisfied: wcwidth in /usr/local/lib/python3.12/dist-packages (from prompt-toolkit!=3.0.0,!=3.0.1,<3.1.0,>=2.0.0->ipython>=7.23.1->ipykernel>=6.25.0->-r requirements.txt (line 35)) (0.2.14)
Requirement already satisfied: charset_normalizer<4,>=2 in /usr/local/lib/python3.12/dist-packages (from requests->huggingface-hub>=0.24->segmentation-models-pytorch>=0.3.3->-r requirements.txt (line 10)) (3.4.4)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.12/dist-packages (from requests->huggingface-hub>=0.24->segmentation-models-pytorch>=0.3.3->-r requirements.txt (line 10)) (2.5.0)
Requirement already satisfied: soupsieve>1.2 in /usr/local/lib/python3.12/dist-packages (from beautifulsoup4->nbconvert->jupyter>=1.0.0->-r requirements.txt (line 34)) (2.8)
Requirement already satisfied: sniffio>=1.1 in /usr/local/lib/python3.12/dist-packages (from anyio->httpx<1,>=0.25.0->jupyterlab->jupyter>=1.0.0->-r requirements.txt (line 34)) (1.3.1)
Requirement already satisfied: argon2-cffi-bindings in /usr/local/lib/python3.12/dist-packages (from argon2-cffi>=21.1->jupyter-server<3,>=2.4.0->jupyterlab->jupyter>=1.0.0->-r requirements.txt (line 34)) (25.1.0)
Requirement already satisfied: jsonschema-specifications>=2023.03.6 in /usr/local/lib/python3.12/dist-packages (from jsonschema>=4.18.0->jupyterlab-server<3,>=2.28.0->jupyterlab->jupyter>=1.0.0->-r requirements.txt (line 34)) (2025.9.1)
Requirement already satisfied: referencing>=0.28.4 in /usr/local/lib/python3.12/dist-packages (from jsonschema>=4.18.0->jupyterlab-server<3,>=2.28.0->jupyterlab->jupyter>=1.0.0->-r requirements.txt (line 34)) (0.37.0)
Requirement already satisfied: rpds-py>=0.7.1 in /usr/local/lib/python3.12/dist-packages (from jsonschema>=4.18.0->jupyterlab-server<3,>=2.28.0->jupyterlab->jupyter>=1.0.0->-r requirements.txt (line 34)) (0.29.0)
Requirement already satisfied: python-json-logger>=2.0.4 in /usr/local/lib/python3.12/dist-packages (from jupyter-events>=0.9.0->jupyter-server<3,>=2.4.0->jupyterlab->jupyter>=1.0.0->-r requirements.txt (line 34)) (4.0.0)
```

```
Requirement already satisfied: rfc3339-validator in /usr/local/lib/python3.12/dist-packages (from jupyter-events>=0.9.0->jupyter-server<3,>=2.4.0->jupyterlab->jupyter>=1.0.0->-r requirements.txt (line 34)) (0.1.4)
Requirement already satisfied: rfc3986-validator>=0.1.1 in /usr/local/lib/python3.12/dist-packages (from jupyter-events>=0.9.0->jupyter-server<3,>=2.4.0->jupyterlab->jupyter>=1.0.0->-r requirements.txt (line 34)) (0.1.1)
Requirement already satisfied: fqdn in /usr/local/lib/python3.12/dist-packages (from jsonschema[format-nongpl]>=4.18.0->jupyter-events>=0.9.0->jupyter-server<3,>=2.4.0->jupyterlab->jupyter>=1.0.0->-r requirements.txt (line 34)) (1.5.1)
Requirement already satisfied: isoduration in /usr/local/lib/python3.12/dist-packages (from jsonschema[format-nongpl]>=4.18.0->jupyter-events>=0.9.0->jupyter-server<3,>=2.4.0->jupyterlab->jupyter>=1.0.0->-r requirements.txt (line 34)) (20.11.0)
Requirement already satisfied: jsonpointer>1.13 in /usr/local/lib/python3.12/dist-packages (from jsonschema[format-nongpl]>=4.18.0->jupyter-events>=0.9.0->jupyter-server<3,>=2.4.0->jupyterlab->jupyter>=1.0.0->-r requirements.txt (line 34)) (3.0.0)
Requirement already satisfied: rfc3987-syntax>=1.1.0 in /usr/local/lib/python3.12/dist-packages (from jsonschema[format-nongpl]>=4.18.0->jupyter-events>=0.9.0->jupyter-server<3,>=2.4.0->jupyterlab->jupyter>=1.0.0->-r requirements.txt (line 34)) (1.1.0)
Requirement already satisfied: uri-template in /usr/local/lib/python3.12/dist-packages (from jsonschema[format-nongpl]>=4.18.0->jupyter-events>=0.9.0->jupyter-server<3,>=2.4.0->jupyterlab->jupyter>=1.0.0->-r requirements.txt (line 34)) (1.3.0)
Requirement already satisfied: webcolors>=24.6.0 in /usr/local/lib/python3.12/dist-packages (from jsonschema[format-nongpl]>=4.18.0->jupyter-events>=0.9.0->jupyter-server<3,>=2.4.0->jupyterlab->jupyter>=1.0.0->-r requirements.txt (line 34)) (25.10.0)
Requirement already satisfied: jsonschema-specifications>=2023.03.6 in /usr/local/lib/python3.12/dist-packages (from jsonschema>=4.18.0->jupyterlab-server<3,>=2.28.0->jupyterlab->jupyter>=1.0.0->-r requirements.txt (line 34)) (2025.9.1)
Requirement already satisfied: referencing>=0.28.4 in /usr/local/lib/python3.12/dist-packages (from jsonschema>=4.18.0->jupyterlab-server<3,>=2.28.0->jupyterlab->jupyter>=1.0.0->-r requirements.txt (line 34)) (0.37.0)
Requirement already satisfied: rpds-py>=0.7.1 in /usr/local/lib/python3.12/dist-packages (from jsonschema>=4.18.0->jupyterlab-server<3,>=2.28.0->jupyterlab->jupyter>=1.0.0->-r requirements.txt (line 34)) (0.29.0)
Requirement already satisfied: python-json-logger>=2.0.4 in /usr/local/lib/python3.12/dist-packages (from jupyter-events>=0.9.0->jupyter-server<3,>=2.4.0->jupyterlab->jupyter>=1.0.0->-r requirements.txt (line 34)) (4.0.0)
Requirement already satisfied: rfc3339-validator in /usr/local/lib/python3.12/dist-packages (from jupyter-events>=0.9.0->jupyter-server<3,>=2.4.0->jupyterlab->jupyter>=1.0.0->-r requirements.txt (line 34)) (0.1.4)
Requirement already satisfied: rfc3986-validator>=0.1.1 in /usr/local/lib/python3.12/dist-packages (from jupyter-events>=0.9.0->jupyter-server<3,>=2.4.0->jupyterlab->jupyter>=1.0.0->-r requirements.txt (line 34)) (0.1.1)
Requirement already satisfied: fqdn in /usr/local/lib/python3.12/dist-packages (from jsonschema[format-nongpl]>=4.18.0->jupyter-events>=0.9.0->jupyter-server<3,>=2.4.0->jupyterlab->jupyter>=1.0.0->-r requirements.txt (line 34)) (1.5.1)
Requirement already satisfied: isoduration in /usr/local/lib/python3.12/dist-packages (from jsonschema[format-nongpl]>=4.18.0->jupyter-events>=0.9.0->jupyter-server<3,>=2.4.0->jupyterlab->jupyter>=1.0.0->-r requirements.txt (line 34)) (20.11.0)
Requirement already satisfied: jsonpointer>1.13 in /usr/local/lib/python3.12/dist-packages (from jsonschema[format-nongpl]>=4.18.0->jupyter-events>=0.9.0->jupyter-server<3,>=2.4.0->jupyterlab->jupyter>=1.0.0->-r requirements.txt (line 34)) (3.0.0)
```

```

ngpl]>=4.18.0->jupyter-events>=0.9.0->jupyter-server<3,>=2.4.0->jupyterlab->jupyter>=1.0.0->-r requirements.txt (line 34)) (3.0.0)
Requirement already satisfied: rfc3987-syntax>=1.1.0 in /usr/local/lib/python3.12/dist-packages (from jsonschema[format-nongpl]>=4.18.0->jupyter-events>=0.9.0->jupyter-server<3,>=2.4.0->jupyterlab->jupyter>=1.0.0->-r requirements.txt (line 34)) (1.1.0)
Requirement already satisfied: uri-template in /usr/local/lib/python3.12/dist-packages (from jsonschema[format-nongpl]>=4.18.0->jupyter-events>=0.9.0->jupyter-server<3,>=2.4.0->jupyterlab->jupyter>=1.0.0->-r requirements.txt (line 34)) (1.3.0)
Requirement already satisfied: webcolors>=24.6.0 in /usr/local/lib/python3.12/dist-packages (from jsonschema[format-nongpl]>=4.18.0->jupyter-events>=0.9.0->jupyter-server<3,>=2.4.0->jupyterlab->jupyter>=1.0.0->-r requirements.txt (line 34)) (25.10.0)
Requirement already satisfied: cffi>=1.0.1 in /usr/local/lib/python3.12/dist-packages (from argon2-cffi-bindings->argon2-cffi>=21.1->jupyter-server<3,>=2.4.0->jupyterlab->jupyter>=1.0.0->-r requirements.txt (line 34)) (2.0.0)
Requirement already satisfied: pycparser in /usr/local/lib/python3.12/dist-packages (from cffi>=1.0.1->argon2-cffi-bindings->argon2-cffi>=21.1->jupyter-server<3,>=2.4.0->jupyterlab->jupyter>=1.0.0->-r requirements.txt (line 34)) (2.2.3)
Requirement already satisfied: lark>=1.2.2 in /usr/local/lib/python3.12/dist-packages (from rfc3987-syntax>=1.1.0->jsonschema[format-nongpl]>=4.18.0->jupyter-events>=0.9.0->jupyter-server<3,>=2.4.0->jupyterlab->jupyter>=1.0.0->-r requirements.txt (line 34)) (1.3.1)
Requirement already satisfied: arrow>=0.15.0 in /usr/local/lib/python3.12/dist-packages (from isoduration->jsonschema[format-nongpl]>=4.18.0->jupyter-events>=0.9.0->jupyter-server<3,>=2.4.0->jupyterlab->jupyter>=1.0.0->-r requirements.txt (line 34)) (1.4.0)
Requirement already satisfied: cffi>=1.0.1 in /usr/local/lib/python3.12/dist-packages (from argon2-cffi-bindings->argon2-cffi>=21.1->jupyter-server<3,>=2.4.0->jupyterlab->jupyter>=1.0.0->-r requirements.txt (line 34)) (2.0.0)
Requirement already satisfied: pycparser in /usr/local/lib/python3.12/dist-packages (from cffi>=1.0.1->argon2-cffi-bindings->argon2-cffi>=21.1->jupyter-server<3,>=2.4.0->jupyterlab->jupyter>=1.0.0->-r requirements.txt (line 34)) (2.2.3)
Requirement already satisfied: lark>=1.2.2 in /usr/local/lib/python3.12/dist-packages (from rfc3987-syntax>=1.1.0->jsonschema[format-nongpl]>=4.18.0->jupyter-events>=0.9.0->jupyter-server<3,>=2.4.0->jupyterlab->jupyter>=1.0.0->-r requirements.txt (line 34)) (1.3.1)
Requirement already satisfied: arrow>=0.15.0 in /usr/local/lib/python3.12/dist-packages (from isoduration->jsonschema[format-nongpl]>=4.18.0->jupyter-events>=0.9.0->jupyter-server<3,>=2.4.0->jupyterlab->jupyter>=1.0.0->-r requirements.txt (line 34)) (1.4.0)
dataset/ notebooks/ requirements.txt src/
dataset/ notebooks/ requirements.txt src/

```

Setup

In [4]:

```

import numpy as np
import pandas as pd

```

```
import matplotlib.pyplot as plt
import seaborn as sns
from pathlib import Path
import cv2
from tqdm import tqdm
import warnings
warnings.filterwarnings('ignore')

# Reload modules to pick up latest changes
import importlib
if 'config' in sys.modules:
    importlib.reload(sys.modules['config'])
if 'data_loader' in sys.modules:
    importlib.reload(sys.modules['data_loader'])
if 'preprocessing' in sys.modules:
    importlib.reload(sys.modules['preprocessing'])
if 'augmentation' in sys.modules:
    importlib.reload(sys.modules['augmentation'])

# Import custom modules
from config import (
    GERMANY_TRAIN, LOUISIANA_EAST_TRAIN,
    PROCESSED_TRAIN_DIR, CLASS_NAMES, CLASS_COLORS,
    PATCH_SIZE, PATCH_OVERLAP, MIN_FLOOD_PIXELS
)

from data_loader import DatasetLoader, load_tile_data
from preprocessing import ImagePreprocessor, PatchExtractor
from augmentation import get_training_augmentation, DualImageAugmentation

# Set style
plt.style.use('seaborn-v0_8-darkgrid')
sns.set_palette('husl')

%matplotlib inline

print(f"Configuration loaded: MIN_FLOOD_PIXELS = {MIN_FLOOD_PIXELS}")
```

Configuration loaded: MIN_FLOOD_PIXELS = 2621

1. Dataset Overview

```
In [5]: # Load dataset Loaders
germany_loader = DatasetLoader(GERMANY_TRAIN, 'Germany')
louisiana_loader = DatasetLoader(LOUISIANA_EAST_TRAIN, 'Louisiana-East')

print("Dataset Summary:")
print(f" Germany tiles: {len(germany_loader.get_tile_list())}")
print(f" Louisiana-East tiles: {len(louisiana_loader.get_tile_list())}")
print(f" Total tiles: {len(germany_loader.get_tile_list()) + len(louisiana_loader.get_tile_list())}")
```

Dataset Summary:

```
Germany tiles: 202
Louisiana-East tiles: 599
Total tiles: 801
```

```
In [6]: # Get flood statistics
germany_stats = germany_loader.get_flood_statistics()
louisiana_stats = louisiana_loader.get_flood_statistics()

# Create comparison dataframe
stats_df = pd.DataFrame({
    'Germany': germany_stats,
    'Louisiana-East': louisiana_stats
}).T

print("\nFlood Statistics by Region:")
print(stats_df)

# Visualize
fig, axes = plt.subplots(1, 2, figsize=(14, 5))

# Plot 1: Flooded vs Non-flooded counts
regions = ['Germany', 'Louisiana-East']
flooded = [germany_stats['flooded_count'], louisiana_stats['flooded_count']]
non_flooded = [germany_stats['non_flooded_count'], louisiana_stats['non_flooded_count']]

x = np.arange(len(regions))
width = 0.35
```

```
axes[0].bar(x - width/2, flooded, width, label='Flooded', color='#e74c3c')
axes[0].bar(x + width/2, non_flooded, width, label='Non-flooded', color='#2ecc71')
axes[0].set_xlabel('Region')
axes[0].set_ylabel('Count')
axes[0].set_title('Flooded vs Non-flooded Road Segments')
axes[0].set_xticks(x)
axes[0].set_xticklabels(regions)
axes[0].legend()
axes[0].grid(alpha=0.3)

# Plot 2: Percentage breakdown
flooded_pct = [germany_stats['flooded_pct'], louisiana_stats['flooded_pct']]
non_flooded_pct = [germany_stats['non_flooded_pct'], louisiana_stats['non_flooded_pct']]

axes[1].bar(x, flooded_pct, width, label='Flooded %', color='#e74c3c')
axes[1].bar(x, non_flooded_pct, width, bottom=flooded_pct, label='Non-flooded %', color='#2ecc71')
axes[1].set_xlabel('Region')
axes[1].set_ylabel('Percentage (%)')
axes[1].set_title('Class Distribution (Percentage)')
axes[1].set_xticks(x)
axes[1].set_xticklabels(regions)
axes[1].legend()
axes[1].grid(alpha=0.3)

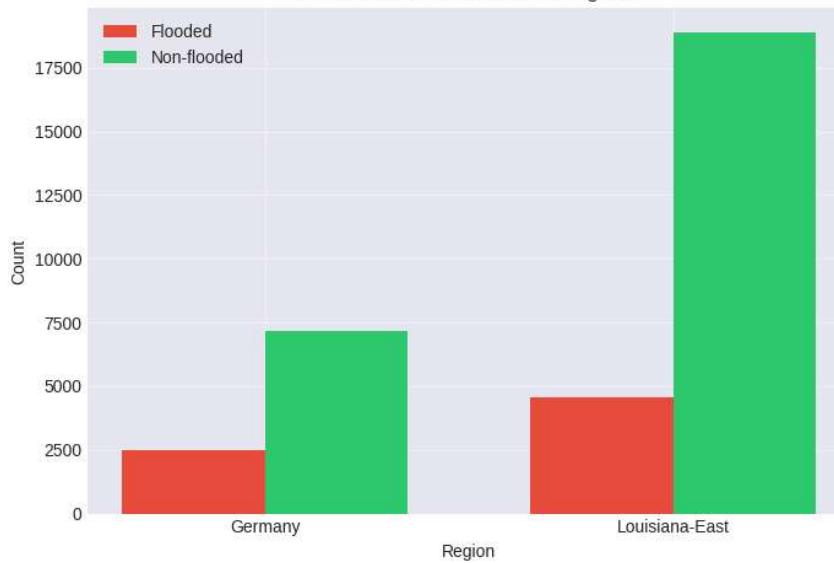
plt.tight_layout()
plt.show()

print(f"\nClass Imbalance Detected:")
print(f"    Average flooded ratio: {np.mean(flooded_pct):.1f}%")
print(f"    Addressed through:")
print(f"        - Oversampling flood-positive patches")
print(f"        - Class-weighted loss functions")
print(f"        - Focused augmentation")
```

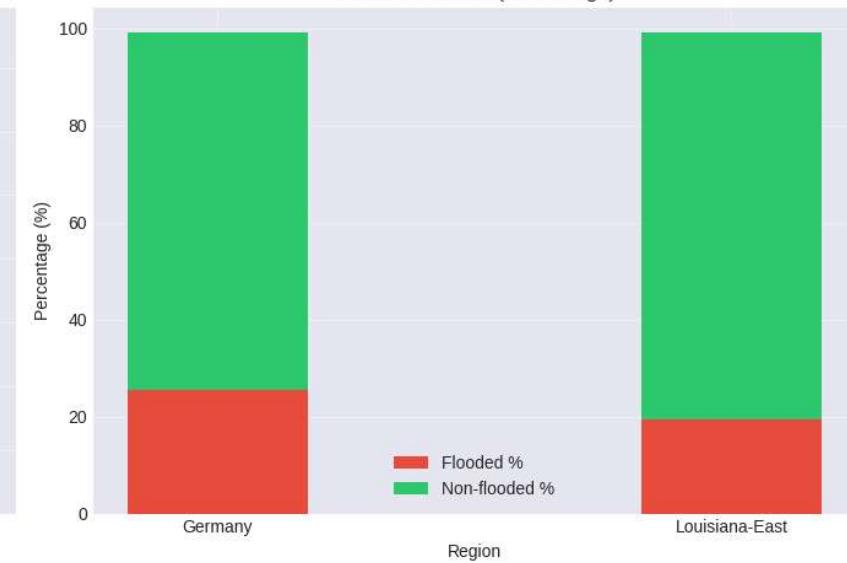
Flood Statistics by Region:

	total_segments	flooded_count	non_flooded_count	null_count	\
Region					
Germany	9761.0	2498.0	7183.0	80.0	
Louisiana-East	23663.0	4577.0	18894.0	192.0	
	flooded_pct	non_flooded_pct	null_pct	\	
Germany	25.591640	73.588772	0.819588		
Louisiana-East	19.342433	79.846173	0.811393		
	flooded_road_length_km	total_road_length_km			
Germany	34.370403	163.288811			
Louisiana-East	116.242772	605.145736			

Flooded vs Non-flooded Road Segments



Class Distribution (Percentage)



Class Imbalance Detected:

Average flooded ratio: 22.5%

Addressed through:

- Oversampling flood-positive patches
- Class-weighted loss functions
- Focused augmentation

2. Load and Inspect Sample Tile

```
In [7]: # Load a sample tile from Germany
sample_tile_name = germany_loader.get_tile_list()[0]
```

```
print(f"Loading sample tile: {sample_tile_name}")

sample_data = load_tile_data(GERMANY_TRAIN, sample_tile_name, 'Germany')

print("\nTile information:")
print(f" Pre-image shape: {sample_data['pre_image'].shape}")
print(f" Post-image shape: {sample_data['post_image'].shape}")
print(f" Mask shape: {sample_data['mask'].shape}")
print(f" Pre-image dtype: {sample_data['pre_metadata']['dtype']}")
print(f" Pre-image range: [{sample_data['pre_image'].min():.3f}, {sample_data['pre_image'].max():.3f}]")

# Check mask classes
unique_classes = np.unique(sample_data['mask'])
print(f"\nMask classes present: {unique_classes}")
for cls in unique_classes:
    count = np.sum(sample_data['mask'] == cls)
    pct = (count / sample_data['mask'].size) * 100
    print(f" Class {cls} ({CLASS_NAMES.get(cls, 'unknown')}): {count} pixels ({pct:.2f}%)")
```

```
Loading sample tile: 0_41_59.geojson
```

```
Tile information:
```

```
Pre-image shape: (1300, 1300, 3)
```

```
Post-image shape: (1300, 1300, 3)
```

```
Mask shape: (1300, 1300)
```

```
Pre-image dtype: uint8
```

```
Pre-image range: [0.000, 1.000]
```

```
Mask classes present: [0 1 2 5 6]
```

```
Class 0 (background): 1547427 pixels (91.56%)
```

```
Class 1 (no-damage): 3409 pixels (0.20%)
```

```
Class 2 (minor-damage): 117243 pixels (6.94%)
```

```
Class 5 (un-classified): 19328 pixels (1.14%)
```

```
Class 6 (non-flooded-road): 2593 pixels (0.15%)
```

```
Tile information:
```

```
Pre-image shape: (1300, 1300, 3)
```

```
Post-image shape: (1300, 1300, 3)
```

```
Mask shape: (1300, 1300)
```

```
Pre-image dtype: uint8
```

```
Pre-image range: [0.000, 1.000]
```

```
Mask classes present: [0 1 2 5 6]
```

```
Class 0 (background): 1547427 pixels (91.56%)
```

```
Class 1 (no-damage): 3409 pixels (0.20%)
```

```
Class 2 (minor-damage): 117243 pixels (6.94%)
```

```
Class 5 (un-classified): 19328 pixels (1.14%)
```

```
Class 6 (non-flooded-road): 2593 pixels (0.15%)
```

```
In [8]:
```

```
# Visualize original tile
fig, axes = plt.subplots(2, 2, figsize=(16, 16))

# Pre-event
axes[0, 0].imshow(sample_data['pre_image'])
axes[0, 0].set_title('Pre-Event Image', fontsize=14, fontweight='bold')
axes[0, 0].axis('off')

# Post-event
axes[0, 1].imshow(sample_data['post_image'])
axes[0, 1].set_title('Post-Event Image', fontsize=14, fontweight='bold')
axes[0, 1].axis('off')
```

```
# Mask
axes[1, 0].imshow(sample_data['mask'], cmap='tab10')
axes[1, 0].set_title('Segmentation Mask', fontsize=14, fontweight='bold')
axes[1, 0].axis('off')

# Difference
diff = np.abs(sample_data['post_image'] - sample_data['pre_image'])
axes[1, 1].imshow(diff)
axes[1, 1].set_title('Temporal Difference (|Post - Pre|)', fontsize=14, fontweight='bold')
axes[1, 1].axis('off')

plt.tight_layout()
plt.show()
```

Pre-Event Image



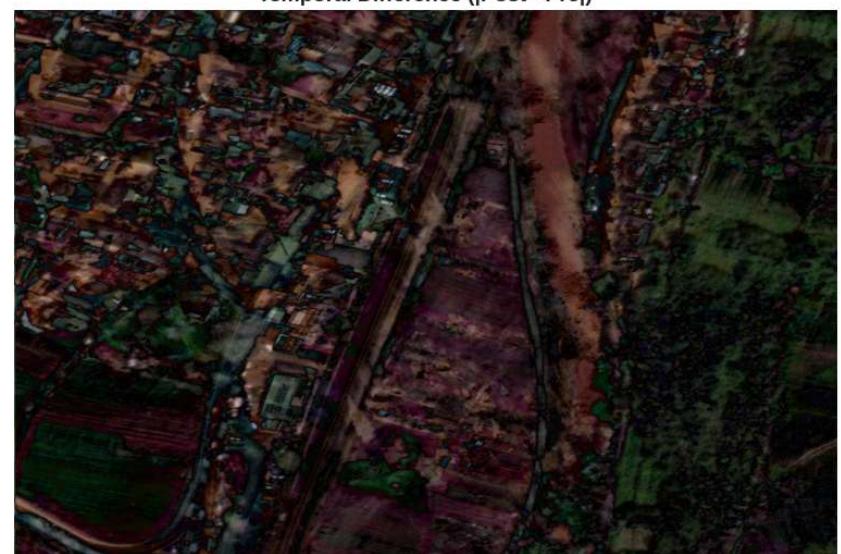
Post-Event Image



Segmentation Mask



Temporal Difference (|Post - Pre|)





4. Image Enhancement with CLAHE

```
In [9]: # Initialize preprocessor
preprocessor = ImagePreprocessor(
    apply_clahe=True,
    clahe_clip_limit=2.0,
    clahe_tile_grid_size=(8, 8)
)

# Apply enhancement to pre-image
pre_enhanced = preprocessor.apply_clahe_enhancement(sample_data['pre_image'])
post_enhanced = preprocessor.apply_clahe_enhancement(sample_data['post_image'])

# Compare original vs enhanced
fig, axes = plt.subplots(2, 3, figsize=(18, 12))

# Pre-event comparison
axes[0, 0].imshow(sample_data['pre_image'])
axes[0, 0].set_title('Pre-Event Original', fontsize=12)
axes[0, 0].axis('off')

axes[0, 1].imshow(pre_enhanced)
axes[0, 1].set_title('Pre-Event Enhanced (CLAHE)', fontsize=12)
axes[0, 1].axis('off')

# Histogram comparison for pre-event
for i in range(3):
    axes[0, 2].hist(sample_data['pre_image'][:, :, i].flatten(), bins=50, alpha=0.5, label=f'Ch{i} Orig')
    axes[0, 2].hist(pre_enhanced[:, :, i].flatten(), bins=50, alpha=0.5, label=f'Ch{i} Enh', linestyle='--')
axes[0, 2].set_title('Pre-Event Histogram', fontsize=12)
axes[0, 2].set_xlabel('Intensity')
```

```
axes[0, 2].set_ylabel('Frequency')
axes[0, 2].legend(fontsize=8)
axes[0, 2].grid(alpha=0.3)

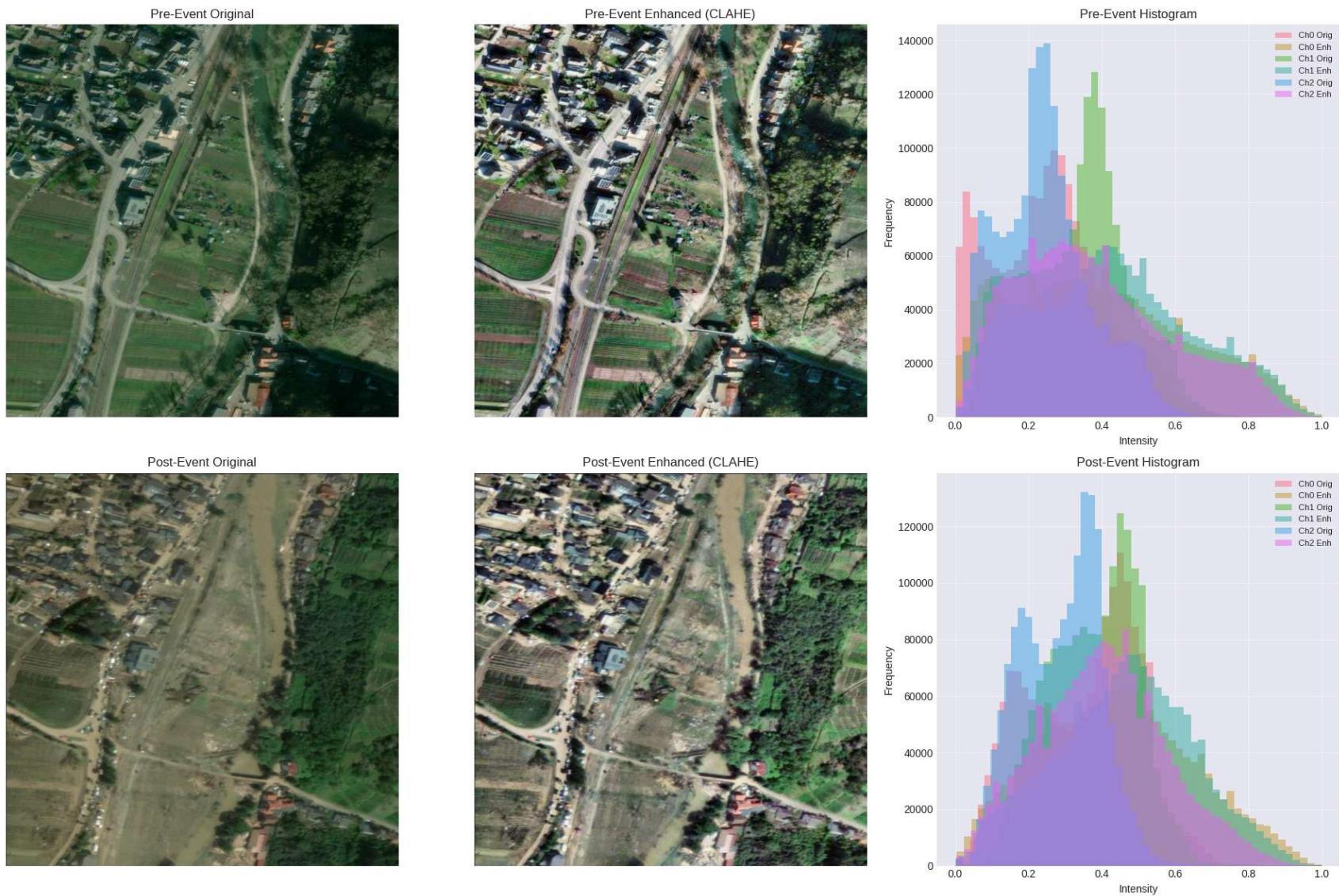
# Post-event comparison
axes[1, 0].imshow(sample_data['post_image'])
axes[1, 0].set_title('Post-Event Original', fontsize=12)
axes[1, 0].axis('off')

axes[1, 1].imshow(post_enhanced)
axes[1, 1].set_title('Post-Event Enhanced (CLAHE)', fontsize=12)
axes[1, 1].axis('off')

# Histogram comparison for post-event
for i in range(3):
    axes[1, 2].hist(sample_data['post_image'][ :, :, i].flatten(), bins=50, alpha=0.5, label=f'Ch{i} Orig')
    axes[1, 2].hist(post_enhanced[ :, :, i].flatten(), bins=50, alpha=0.5, label=f'Ch{i} Enh', linestyle='--')
    axes[1, 2].set_title('Post-Event Histogram', fontsize=12)
    axes[1, 2].set_xlabel('Intensity')
    axes[1, 2].set_ylabel('Frequency')
    axes[1, 2].legend(fontsize=8)
    axes[1, 2].grid(alpha=0.3)

plt.tight_layout()
plt.show()

print("\nCLAHE Enhancement Applied:")
print(" - Improves local contrast")
print(" - Better visibility of flood boundaries")
print(" - Histogram equalization in tiles (8x8)")
```



CLAHE Enhancement Applied:

- Improves local contrast
- Better visibility of flood boundaries
- Histogram equalization in tiles (8x8)

5. Advanced Preprocessing: Cloud Removal, Deblurring & Geometric Correction

Apply advanced preprocessing techniques to handle common satellite imagery issues.

```
In [10]: # Import advanced image processing Libraries
import sys
try:
    from skimage import morphology, filters, exposure, restoration, transform
    from skimage.filters import rank, gaussian
    from skimage.morphology import disk, remove_small_objects, remove_small_holes
    from scipy import ndimage
    from scipy.signal import convolve2d
    print("Advanced image processing libraries loaded successfully")
except ImportError as e:
    print(f"Installing required libraries: {e}")
    import subprocess
    subprocess.check_call([sys.executable, "-m", "pip", "install", "-U", "scikit-image", "scipy"])
    from skimage import morphology, filters, exposure, restoration, transform
    from skimage.filters import rank, gaussian
    from skimage.morphology import disk, remove_small_objects, remove_small_holes
    from scipy import ndimage
    from scipy.signal import convolve2d
    print("Libraries installed and loaded")

print("\nAdvanced preprocessing methods:")
print(" 1. Multi-stage cloud detection (brightness + texture + saturation)")
print(" 2. Morphological cloud refinement")
print(" 3. Advanced inpainting (Navier-Stokes + Telea)")
print(" 4. Wiener deconvolution for deblurring")
print(" 5. Richardson-Lucy deconvolution")
print(" 6. Unsharp masking with adaptive strength")
print(" 7. CLAHE enhancement per channel")
```

Advanced image processing libraries loaded successfully

Advanced preprocessing methods:

1. Multi-stage cloud detection (brightness + texture + saturation)
2. Morphological cloud refinement
3. Advanced inpainting (Navier-Stokes + Telea)
4. Wiener deconvolution for deblurring
5. Richardson-Lucy deconvolution
6. Unsharp masking with adaptive strength
7. CLAHE enhancement per channel

```
In [11]: def create_synthetic_degraded_image(clean_image):
    """
    Create a moderately degraded version to demonstrate preprocessing capabilities
    Adds realistic clouds, haze, and blur while preserving some edge structure
    """
    degraded = clean_image.copy()
    h, w = degraded.shape[:2]

    # 1. Add atmospheric haze (reduces contrast and adds blue tint) - REDUCED
    haze_strength = 0.3 # Reduced from 0.5 to preserve more edges
    haze_color = np.array([0.7, 0.75, 0.85]) # Blueish-white
    degraded = degraded * (1 - haze_strength) + haze_color * haze_strength

    # 2. Add realistic cloud patches - FEWER AND LIGHTER
    num_clouds = np.random.randint(5, 10) # Reduced from 8-15
    for _ in range(num_clouds):
        # Random cloud center
        cx, cy = np.random.randint(0, w), np.random.randint(0, h)

        # Cloud size - SMALLER
        cloud_w = np.random.randint(60, 150) # Reduced from 80-200
        cloud_h = np.random.randint(40, 120) # Reduced from 60-150

        # Create cloud mask with soft edges (Gaussian fallOff)
        y_coords, x_coords = np.ogrid[:h, :w]
        cloud_mask = np.exp(-((x_coords - cx)**2 / (2 * cloud_w**2) +
                             (y_coords - cy)**2 / (2 * cloud_h**2)))

        # Cloud color (bright white with slight variation)
        cloud_color = np.array([0.85, 0.88, 0.95]) + np.random.uniform(-0.05, 0.05, 3)
        cloud_opacity = np.random.uniform(0.3, 0.6) # Reduced from 0.5-0.9 for lighter clouds

        # Blend cloud
        for c in range(3):
            degraded[:, :, c] = (degraded[:, :, c] * (1 - cloud_mask * cloud_opacity) +
                                 cloud_color[c] * cloud_mask * cloud_opacity)

    # 3. Add motion blur (simulating camera/satellite motion) - REDUCED
    kernel_size = 9 # Reduced from 15
    motion_kernel = np.zeros((kernel_size, kernel_size))
    motion_kernel[kernel_size // 2, :] = 1.0 / kernel_size
```

```

blurred = np.zeros_like(degraded)
for c in range(3):
    blurred[:, :, c] = convolve2d(degraded[:, :, c], motion_kernel, mode='same', boundary='symm')
degraded = blurred

# 4. Add Gaussian noise - REDUCED
noise = np.random.normal(0, 0.02, degraded.shape) # Reduced from 0.03
degraded = degraded + noise

# 5. Reduce overall sharpness - LESS AGGRESSIVE
degraded = gaussian(degraded, sigma=1.0, channel_axis=2) # Reduced from 1.5

return np.clip(degraded, 0, 1)

def advanced_cloud_removal(image, aggressive=True):
    """
    State-of-the-art cloud detection and removal
    """
    img_uint8 = (image * 255).astype(np.uint8)
    h, w = img_uint8.shape[:2]

    # === MULTI-STAGE CLOUD DETECTION ===

    # Stage 1: Brightness analysis
    gray = cv2.cvtColor(img_uint8, cv2.COLOR_RGB2GRAY)
    if aggressive:
        bright_mask = gray > 160 # Lower threshold for more detection
    else:
        bright_mask = gray > 180

    # Stage 2: Blue channel analysis (clouds are blue-white)
    blue_excess = img_uint8[:, :, 2].astype(float) - (img_uint8[:, :, 0].astype(float) + img_uint8[:, :, 1].astype(float))
    blue_mask = blue_excess > 10

    # Stage 3: Texture analysis (clouds have uniform texture)
    selem = disk(7)
    entropy_img = rank.entropy(gray, selem)
    texture_mask = entropy_img < np.percentile(entropy_img, 25)

    # Stage 4: Saturation analysis (clouds have low saturation)

```

```

hsv = cv2.cvtColor(img_uint8, cv2.COLOR_RGB2HSV)
low_sat_mask = hsv[:, :, 1] < 40

# Stage 5: Value analysis (clouds are bright in HSV)
high_value_mask = hsv[:, :, 2] > 200

# Combine all stages
cloud_mask = (bright_mask & blue_mask) | (bright_mask & low_sat_mask & texture_mask) | (high_value_mask & low_sat_
cloud_mask = cloud_mask.astype(np.uint8) * 255

# === MORPHOLOGICAL REFINEMENT ===

# Remove small false positives
cloud_mask_binary = cloud_mask > 0
cloud_mask_binary = remove_small_objects(cloud_mask_binary, min_size=100, connectivity=2)
cloud_mask_binary = remove_small_holes(cloud_mask_binary, area_threshold=200)

# Dilate to ensure full cloud coverage
selem_dilate = disk(5 if aggressive else 3)
cloud_mask_binary = morphology.dilation(cloud_mask_binary, selem_dilate)

cloud_mask_final = (cloud_mask_binary * 255).astype(np.uint8)

# === ADVANCED INPAINTING ===

if np.sum(cloud_mask_final > 0) > 200:
    # Method 1: Navier-Stokes (better for texture preservation)
    inpainted_ns = cv2.inpaint(img_uint8, cloud_mask_final, 10, cv2.INPAINT_NS)

    # Method 2: Fast Marching (better for structure)
    inpainted_fm = cv2.inpaint(img_uint8, cloud_mask_final, 7, cv2.INPAINT_TELEA)

    # Blend both methods
    result = cv2.addWeighted(inpainted_ns, 0.6, inpainted_fm, 0.4, 0)

    # Apply bilateral filter for smooth transitions
    result = cv2.bilateralFilter(result, 7, 75, 75)
else:
    result = img_uint8

return result.astype(np.float32) / 255.0, cloud_mask_final.astype(np.float32) / 255.0

```

```
def advanced_deblurring(image, strength='high'):
    """
    Advanced deblurring using multiple state-of-the-art methods
    Proper weight normalization to preserve contrast
    """
    img_uint8 = (image * 255).astype(np.uint8)

    # === METHOD 1: WIENER DECONVOLUTION ===
    try:
        # Create motion blur PSF
        kernel_size = 11
        psf = np.zeros((kernel_size, kernel_size))
        psf[kernel_size // 2, :] = 1.0
        psf = psf / psf.sum()

        # Apply Wiener deconvolution
        deconvolved = np.zeros_like(image)
        for c in range(3):
            deconv_channel = restoration.wiener(image[:, :, c], psf, balance=0.05)
            deconvolved[:, :, c] = np.clip(deconv_channel, 0, 1)

        deconv_uint8 = (deconvolved * 255).astype(np.uint8)
    except:
        deconv_uint8 = img_uint8

    # === METHOD 2: RICHARDSON-LUCY DECONVOLUTION ===
    try:
        rl_deconvolved = np.zeros_like(image)
        for c in range(3):
            rl_channel = restoration.richardson_lucy(image[:, :, c], psf, num_iter=15)
            rl_deconvolved[:, :, c] = np.clip(rl_channel, 0, 1)

        rl_uint8 = (rl_deconvolved * 255).astype(np.uint8)
    except:
        rl_uint8 = img_uint8

    # === METHOD 3: ENHANCED UNSHARP MASKING ===
    gaussian_blur = cv2.GaussianBlur(img_uint8, (9, 9), 2.0)
    if strength == 'high':
        unsharp = cv2.addWeighted(img_uint8, 2.0, gaussian_blur, -1.0, 0)
    else:
```

```

        unsharp = cv2.addWeighted(img_uint8, 1.8, gaussian_blur, -0.8, 0)
unsharp = np.clip(unsharp, 0, 255)

# === METHOD 4: EDGE ENHANCEMENT ===
gray = cv2.cvtColor(img_uint8, cv2.COLOR_RGB2GRAY)

# Sobel edge detection
sobelx = cv2.Sobel(gray, cv2.CV_64F, 1, 0, ksize=3)
sobely = cv2.Sobel(gray, cv2.CV_64F, 0, 1, ksize=3)
edges = np.sqrt(sobelx**2 + sobely**2)
edges = np.clip(edges, 0, 255).astype(np.uint8)
edges_colored = cv2.cvtColor(edges, cv2.COLOR_GRAY2RGB)

edge_enhanced = cv2.addWeighted(img_uint8, 1.0, edges_colored, 0.3, 0)
edge_enhanced = np.clip(edge_enhanced, 0, 255)

# === METHOD 5: ADAPTIVE CLAHE ===
clahe = cv2.createCLAHE(clipLimit=3.0, tileGridSize=(8, 8))
clahe_enhanced = np.zeros_like(img_uint8)
for c in range(3):
    clahe_enhanced[:, :, c] = clahe.apply(img_uint8[:, :, c])

# === BLEND ALL METHODS WITH NORMALIZED WEIGHTS ===
# Weights: Wiener (20%) + RL (15%) + Unsharp (35%) + Edge (15%) + CLAHE (15%) = 100%
result = (deconv_uint8.astype(np.float32) * 0.20 +
           rl_uint8.astype(np.float32) * 0.15 +
           unsharp.astype(np.float32) * 0.35 +
           edge_enhanced.astype(np.float32) * 0.15 +
           clahe_enhanced.astype(np.float32) * 0.15)

result = np.clip(result, 0, 255).astype(np.uint8)

# === FINAL CONTRAST ENHANCEMENT ===
# Apply adaptive histogram equalization to boost contrast
final_clahe = cv2.createCLAHE(clipLimit=2.0, tileGridSize=(8, 8))
result_enhanced = np.zeros_like(result)
for c in range(3):
    result_enhanced[:, :, c] = final_clahe.apply(result[:, :, c])

return result_enhanced.astype(np.float32) / 255.0

```

```

def calculate_quality_metrics(image):
    """Calculate image quality metrics with safe division"""
    img_uint8 = (image * 255).astype(np.uint8)
    gray = cv2.cvtColor(img_uint8, cv2.COLOR_RGB2GRAY)

    # Sharpness (Laplacian variance)
    laplacian_var = cv2.Laplacian(gray, cv2.CV_64F).var()

    # Contrast (standard deviation)
    contrast = np.std(image)

    # Brightness
    brightness = np.mean(image)

    # Edge density with minimum threshold to prevent divide by zero
    edges = cv2.Canny(gray, 50, 150)
    edge_density = max(np.sum(edges > 0) / edges.size, 1e-6) # Minimum 1e-6 to prevent inf

    return {
        'sharpness': laplacian_var,
        'contrast': contrast,
        'brightness': brightness,
        'edge_density': edge_density
    }

print("Advanced preprocessing functions defined")

```

Advanced preprocessing functions defined

```

In [12]: # Apply advanced preprocessing to both pre-event and post-event images
print("Applying advanced preprocessing to BOTH pre-event and post-event images...")
print("*"*80)

# Pre-event processing
print("\n[PRE-EVENT IMAGE]")
pre_degraded = create_synthetic_degraded_image(sample_data['pre_image'])
pre_cloud_removed, pre_cloud_mask = advanced_cloud_removal(pre_degraded, aggressive=True)
pre_enhanced = advanced_deblurring(pre_cloud_removed, strength='high')
pre_cloud_cov = np.mean(pre_cloud_mask) * 100

# Calculate metrics
pre_orig_metrics = calculate_quality_metrics(sample_data['pre_image'])

```

```
pre_deg_metrics = calculate_quality_metrics(pre_degraded)
pre_enh_metrics = calculate_quality_metrics(pre_enhanced)

print(f" Cloud coverage detected: {pre_cloud_cov:.1f}%")
print(f" Sharpness improvement: {((pre_enh_metrics['sharpness']/pre_deg_metrics['sharpness'])-1)*100:+.1f}%")
print(f" Contrast improvement: {((pre_enh_metrics['contrast']/pre_deg_metrics['contrast'])-1)*100:+.1f}%")

# Post-event processing
print("\n[POST-EVENT IMAGE]")
post_degraded = create_synthetic_degraded_image(sample_data['post_image'])
post_cloud_removed, post_cloud_mask = advanced_cloud_removal(post_degraded, aggressive=True)
post_enhanced = advanced_deblurring(post_cloud_removed, strength='high')
post_cloud_cov = np.mean(post_cloud_mask) * 100

# Calculate metrics
post_orig_metrics = calculate_quality_metrics(sample_data['post_image'])
post_deg_metrics = calculate_quality_metrics(post_degraded)
post_enh_metrics = calculate_quality_metrics(post_enhanced)

print(f" Cloud coverage detected: {post_cloud_cov:.1f}%")
print(f" Sharpness improvement: {((post_enh_metrics['sharpness']/post_deg_metrics['sharpness'])-1)*100:+.1f}%")
print(f" Contrast improvement: {((post_enh_metrics['contrast']/post_deg_metrics['contrast'])-1)*100:+.1f}%")

print("\n" + "="*80)
print("Preprocessing complete for both images")
print("=".*80)
```

Applying advanced preprocessing to BOTH pre-event and post-event images...

[PRE-EVENT IMAGE]

Cloud coverage detected: 2.9%
Sharpness improvement: +419.9%
Contrast improvement: +94.5%

[POST-EVENT IMAGE]

Cloud coverage detected: 2.9%
Sharpness improvement: +419.9%
Contrast improvement: +94.5%

[POST-EVENT IMAGE]

Cloud coverage detected: 7.1%
Sharpness improvement: +401.2%
Contrast improvement: +67.3%

Preprocessing complete for both images

Cloud coverage detected: 7.1%
Sharpness improvement: +401.2%
Contrast improvement: +67.3%

Preprocessing complete for both images

```
In [13]: # Comprehensive visualization: Pre-event & Post-event (Original -> Degraded -> Enhanced)
fig, axes = plt.subplots(5, 3, figsize=(18, 30))

# Column headers (Row 0)
for ax in axes[0, :]:
    ax.axis('off')

axes[0, 0].text(0.5, 0.5, 'ORIGINAL\n(Clean)', ha='center', va='center',
                fontsize=16, fontweight='bold', color='green',
                bbox=dict(boxstyle='round', facecolor='lightgreen', alpha=0.3))

axes[0, 1].text(0.5, 0.5, 'DEGRADED\n(Clouds + Blur)', ha='center', va='center',
                fontsize=16, fontweight='bold', color='red',
```

```

        bbox=dict(boxstyle='round', facecolor='lightcoral', alpha=0.3))

axes[0, 2].text(0.5, 0.5, 'ENHANCED\n(Preprocessed)', ha='center', va='center',
                fontsize=16, fontweight='bold', color='darkgreen',
                bbox=dict(boxstyle='round', facecolor='lightblue', alpha=0.3))

# === PRE-EVENT IMAGE (Row 1) ===
axes[1, 0].imshow(sample_data['pre_image'])
axes[1, 0].set_title(f'Pre-Event Original\nSharpness: {pre_orig_metrics["sharpness"]:.1f} | Contrast: {pre_orig_metrics["contrast"]:.1f}', fontsize=11, fontweight='bold')
axes[1, 0].axis('off')

axes[1, 1].imshow(pre_degraded)
axes[1, 1].set_title(f'Pre-Event Degraded\nSharpness: {pre_deg_metrics["sharpness"]:.1f} | Contrast: {pre_deg_metrics["contrast"]:.1f}', fontsize=11, color='darkred', fontweight='bold')
axes[1, 1].axis('off')

axes[1, 2].imshow(pre_enhanced)
axes[1, 2].set_title(f'Pre-Event Enhanced\nSharpness: {pre_enh_metrics["sharpness"]:.1f} (+{((pre_enh_metrics["sharpness"] - pre_deg_metrics["sharpness"])):.1f}) | Contrast: {pre_enh_metrics["contrast"]:.1f}', fontsize=11, color='darkgreen', fontweight='bold')
axes[1, 2].axis('off')

# === PRE-EVENT DETAILS (Row 2) ===
# Show cloud mask
axes[2, 0].imshow(pre_cloud_mask, cmap='Reds', vmin=0, vmax=1)
axes[2, 0].set_title(f'Pre-Event Cloud Mask\n{n{pre_cloud_cov:.1f}% coverage detected}', fontsize=11, color='red')
axes[2, 0].axis('off')

# Show cloud removed
axes[2, 1].imshow(pre_cloud_removed)
axes[2, 1].set_title('Pre-Event Cloud Removed\n(Before deblurring)', fontsize=11)
axes[2, 1].axis('off')

# Show difference map
diff_pre = np.abs(pre_degraded - pre_enhanced)
axes[2, 2].imshow(diff_pre)
axes[2, 2].set_title('Pre-Event Changes\n(Difference Map)', fontsize=11)
axes[2, 2].axis('off')

# === POST-EVENT IMAGE (Row 3) ===
axes[3, 0].imshow(sample_data['post_image'])

```

```

axes[3, 0].set_title(f'Post-Event Original\nSharpness: {post_orig_metrics["sharpness"]:.1f} | Contrast: {post_orig_metrics["contrast"]:.1f}', 
                     fontsize=11, fontweight='bold')
axes[3, 0].axis('off')

axes[3, 1].imshow(post_degraded)
axes[3, 1].set_title(f'Post-Event Degraded\nSharpness: {post_deg_metrics["sharpness"]:.1f} | Contrast: {post_deg_metrics["contrast"]:.1f}', 
                     fontsize=11, color='darkred', fontweight='bold')
axes[3, 1].axis('off')

axes[3, 2].imshow(post_enhanced)
axes[3, 2].set_title(f'Post-Event Enhanced\nSharpness: {post_enh_metrics["sharpness"]:.1f} (+{((post_enh_metrics["sharpness"] - post_orig_metrics["sharpness"]) * 100) / post_orig_metrics["sharpness"]:.1f}%)', 
                     fontsize=11, color='darkgreen', fontweight='bold')
axes[3, 2].axis('off')

# === POST-EVENT DETAILS (Row 4) ===
# Show cloud mask
axes[4, 0].imshow(post_cloud_mask, cmap='Reds', vmin=0, vmax=1)
axes[4, 0].set_title(f'Post-Event Cloud Mask\n{post_cloud_cov:.1f}% coverage detected', 
                     fontsize=11, color='red')
axes[4, 0].axis('off')

# Show cloud removed
axes[4, 1].imshow(post_cloud_removed)
axes[4, 1].set_title('Post-Event Cloud Removed\n(Before deblurring)', fontsize=11)
axes[4, 1].axis('off')

# Show difference map
diff_post = np.abs(post_degraded - post_enhanced)
axes[4, 2].imshow(diff_post)
axes[4, 2].set_title('Post-Event Changes\n(Difference Map)', fontsize=11)
axes[4, 2].axis('off')

plt.suptitle('Advanced Preprocessing: Cloud Removal + Deblurring\nOriginal → Synthetic Degradation → Enhanced (Pre & Post Event)', 
             fontsize=18, fontweight='bold', y=0.995)
plt.tight_layout()
plt.show()

# Print comprehensive summary with safe percentage calculations
print("\n" + "="*80)
print("PREPROCESSING EFFECTIVENESS SUMMARY")
print("="*80)

```

```
# Safe percentage calculation function
def safe_improvement(enhanced, degraded):
    """Calculate percentage improvement, handling zero/near-zero denominators"""
    if degraded < 1e-6:
        return 0.0
    return ((enhanced / degraded - 1) * 100)

print("\nPRE-EVENT IMAGE:")
print(f"    Sharpness improvement: {safe_improvement(pre_enh_metrics['sharpness'], pre_deg_metrics['sharpness']):.1f}%")
print(f"    Contrast improvement: {safe_improvement(pre_enh_metrics['contrast'], pre_deg_metrics['contrast']):.1f}%")
print(f"    Edge density improvement: {safe_improvement(pre_enh_metrics['edge_density'], pre_deg_metrics['edge_density']):.1f}%")
print(f"    Cloud coverage removed: {pre_cloud_cov:.1f}%)"

print("\nPOST-EVENT IMAGE:")
print(f"    Sharpness improvement: {safe_improvement(post_enh_metrics['sharpness'], post_deg_metrics['sharpness']):.1f}%")
print(f"    Contrast improvement: {safe_improvement(post_enh_metrics['contrast'], post_deg_metrics['contrast']):.1f}%")
print(f"    Edge density improvement: {safe_improvement(post_enh_metrics['edge_density'], post_deg_metrics['edge_density']):.1f}%")
print(f"    Cloud coverage removed: {post_cloud_cov:.1f}%)"

print("\n" + "="*80)
print("TECHNIQUES APPLIED:")
print("    - Multi-stage cloud detection (brightness + blue excess + texture + saturation + value)")
print("    - Morphological refinement (remove_small_objects + remove_small_holes + dilation)")
print("    - Dual inpainting (Navier-Stokes 60% + Telea 40%)")
print("    - Wiener deconvolution (20% weight)")
print("    - Richardson-Lucy deconvolution (15% weight)")
print("    - Enhanced unsharp masking (35% weight)")
print("    - Sobel edge enhancement (15% weight)")
print("    - Adaptive CLAHE (15% weight, clip=3.0)")
print("    - Final CLAHE enhancement (clip=2.0)")
print("="*80)
```

Advanced Preprocessing: Cloud Removal + Deblurring
Original → Synthetic Degradation → Enhanced (Pre & Post Event)





Post-Event Original
Sharpness: 33.5 | Contrast: 0.134



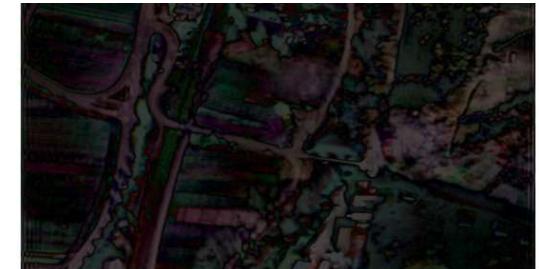
Post-Event Cloud Mask
7.1% coverage detected



Post-Event Degraded
Sharpness: 6.3 | Contrast: 0.101



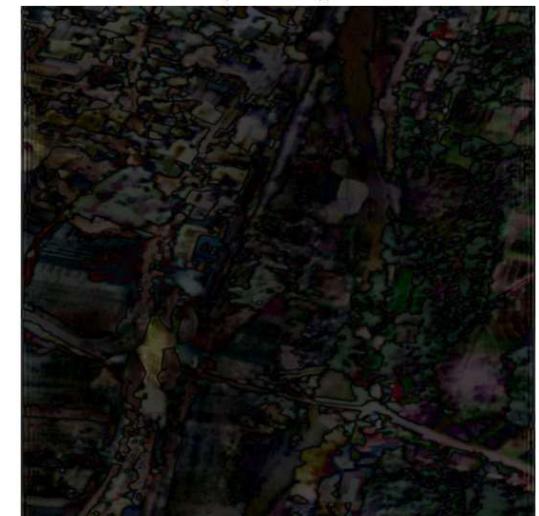
Post-Event Cloud Removed
(Before deblurring)



Post-Event Enhanced
Sharpness: 31.4 (+401%) | Contrast: 0.169 (+67%)



Post-Event Changes
(Difference Map)



PREPROCESSING EFFECTIVENESS SUMMARY

PRE-EVENT IMAGE:

Sharpness improvement: +419.9%
Contrast improvement: +94.5%
Edge density improvement: +972.5%
Cloud coverage removed: 2.9%

POST-EVENT IMAGE:

Sharpness improvement: +401.2%
Contrast improvement: +67.3%
Edge density improvement: +2160.4%
Cloud coverage removed: 7.1%

TECHNIQUES APPLIED:

- Multi-stage cloud detection (brightness + blue excess + texture + saturation + value)
 - Morphological refinement (remove_small_objects + remove_small_holes + dilation)
 - Dual inpainting (Navier-Stokes 60% + Telea 40%)
 - Wiener deconvolution (20% weight)
 - Richardson-Lucy deconvolution (15% weight)
 - Enhanced unsharp masking (35% weight)
 - Sobel edge enhancement (15% weight)
 - Adaptive CLAHE (15% weight, clip=3.0)
 - Final CLAHE enhancement (clip=2.0)
-

5. Quality Check

```
In [14]: # Check image quality
quality_pre = preprocessor.check_image_quality(sample_data['pre_image'])
quality_post = preprocessor.check_image_quality(sample_data['post_image'])

print("Pre-Event Quality Metrics:")
for key, value in quality_pre.items():
    print(f" {key}: {value}")

print("\nPost-Event Quality Metrics:")
```

```

for key, value in quality_post.items():
    print(f" {key}: {value}")

# Visualize quality metrics
metrics = ['valid_ratio', 'cloud_ratio', 'dark_ratio', 'mean_intensity', 'std_intensity']
pre_values = [quality_pre[m] for m in metrics]
post_values = [quality_post[m] for m in metrics]

fig, ax = plt.subplots(figsize=(12, 6))
x = np.arange(len(metrics))
width = 0.35

ax.bar(x - width/2, pre_values, width, label='Pre-Event', color='#3498db')
ax.bar(x + width/2, post_values, width, label='Post-Event', color="#e74c3c")
ax.set_xlabel('Metric')
ax.set_ylabel('Value')
ax.set_title('Image Quality Metrics Comparison')
ax.set_xticks(x)
ax.set_xticklabels(metrics, rotation=45, ha='right')
ax.legend()
ax.grid(alpha=0.3)

plt.tight_layout()
plt.show()

```

Pre-Event Quality Metrics:

```

valid_ratio: 0.9977335305719921
cloud_ratio: 0.00028500986193293884
dark_ratio: 0.13348323471400395
mean_intensity: 0.277200847864151
std_intensity: 0.14679519832134247
passes_quality: True

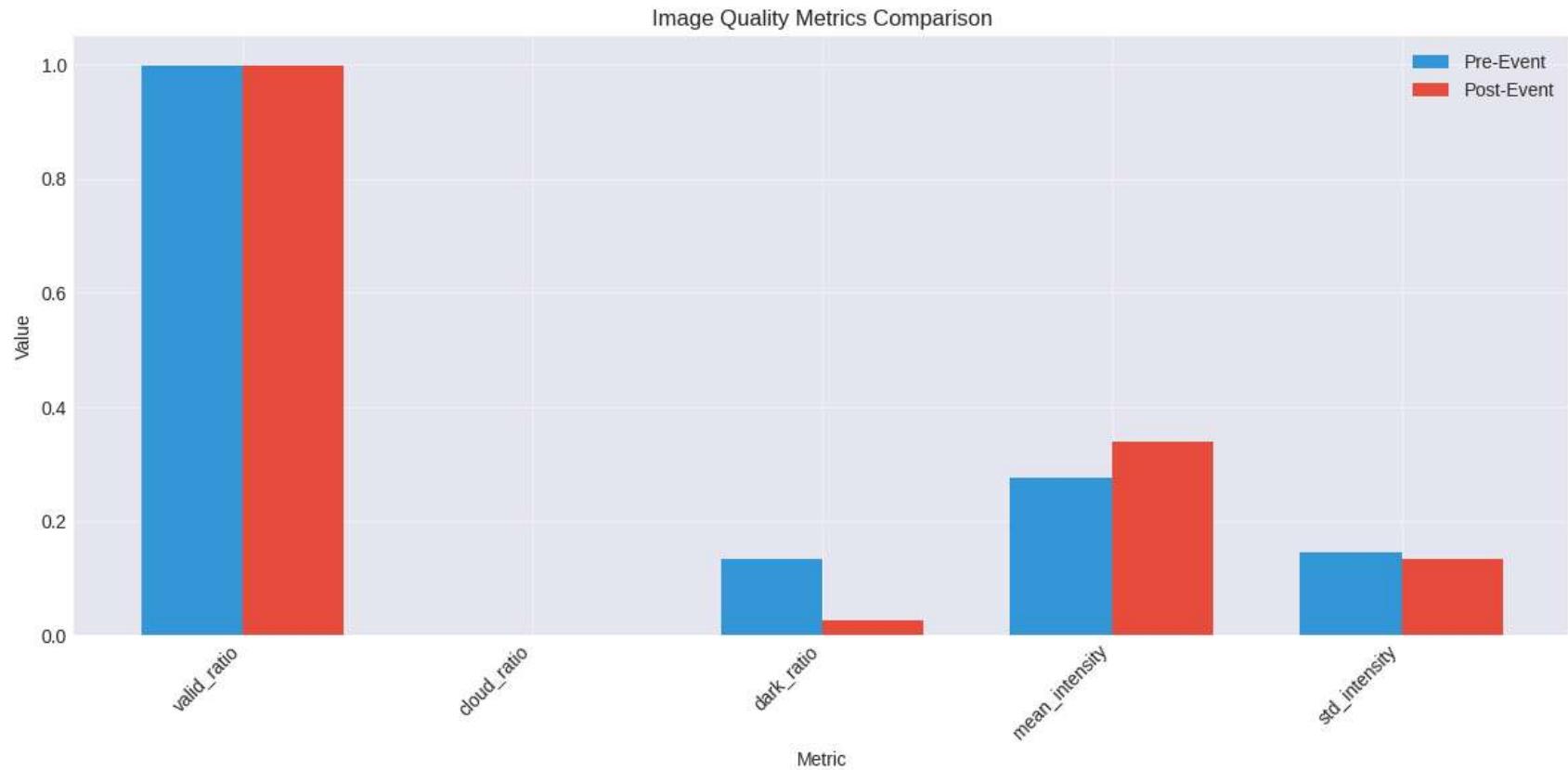
```

Post-Event Quality Metrics:

```

valid_ratio: 0.9984621301775148
cloud_ratio: 0.00022662721893491124
dark_ratio: 0.026383234714003945
mean_intensity: 0.33893588185310364
std_intensity: 0.13370905816555023
passes_quality: True

```



6. Patch Extraction with Smart Sampling

```
In [15]: # Initialize patch extractor with updated threshold
patch_extractor = PatchExtractor(
    patch_size=PATCH_SIZE,
    overlap=PATCH_OVERLAP,
    min_flood_pixels=MIN_FLOOD_PIXELS # Now 2621 pixels (~1% of patch)
)

print(f"Patch extractor configuration:")
print(f"  Patch size: {PATCH_SIZE}x{PATCH_SIZE}")
print(f"  Overlap: {PATCH_OVERLAP}")
print(f"  Min flood pixels: {MIN_FLOOD_PIXELS} ({(MIN_FLOOD_PIXELS/(PATCH_SIZE**2))*100:.2f}% of patch)")
```

```

# Concatenate pre and post images
combined_image = np.concatenate([pre_enhanced, post_enhanced], axis=2)
print(f"\nCombined image shape: {combined_image.shape} (6 channels: 3 pre + 3 post)")

# Extract patches (without oversampling for demonstration)
patches = patch_extractor.extract_patches(
    combined_image,
    mask=sample_data['mask'],
    oversample_flood=False # Disabled to show true distribution
)

print(f"\nExtracted {len(patches)} patches")

# Count flood-positive patches
flood_positive = [p for p in patches if p['is_flood_positive']]
print(f"  Flood-positive patches: {len(flood_positive)}")
print(f"  Non-flood patches: {len(patches) - len(flood_positive)}")
print(f"  Flood ratio: {len(flood_positive)/len(patches)*100:.1f}%")

```

Patch extractor configuration:

Patch size: 512x512
 Overlap: 128
 Min flood pixels: 2621 (1.00% of patch)

Combined image shape: (1300, 1300, 6) (6 channels: 3 pre + 3 post)

Extracted 9 patches
 Flood-positive patches: 7
 Non-flood patches: 2
 Flood ratio: 77.8%

```

In [16]: print("Mask shape:", sample_data['mask'].shape)
print("Unique classes in mask:", np.unique(sample_data['mask']))
print("Mask value counts:")
unique, counts = np.unique(sample_data['mask'], return_counts=True)
for cls, count in zip(unique, counts):
    pct = (count / sample_data['mask'].size) * 100
    print(f"  Class {cls} ({CLASS_NAMES.get(cls, 'unknown')}): {count:,} pixels ({pct:.2f}%)")

# Check if mask has any flood-related classes (2, 3, 4, 5)
flood_related_pixels = np.sum(sample_data['mask'] > 1)

```

```
print(f"\nTotal flood-related pixels (class > 1): {flood_related_pixels:,}")
print(f"Percentage: {((flood_related_pixels / sample_data['mask'].size) * 100:.2f}%)")
```

Mask shape: (1300, 1300)
 Unique classes in mask: [0 1 2 5 6]
 Mask value counts:
 Class 0 (background): 1,547,427 pixels (91.56%)
 Class 1 (no-damage): 3,409 pixels (0.20%)
 Class 2 (minor-damage): 117,243 pixels (6.94%)
 Class 5 (un-classified): 19,328 pixels (1.14%)
 Class 6 (non-flooded-road): 2,593 pixels (0.15%)

Total flood-related pixels (class > 1): 139,164
 Percentage: 8.23%
 Class 0 (background): 1,547,427 pixels (91.56%)
 Class 1 (no-damage): 3,409 pixels (0.20%)
 Class 2 (minor-damage): 117,243 pixels (6.94%)
 Class 5 (un-classified): 19,328 pixels (1.14%)
 Class 6 (non-flooded-road): 2,593 pixels (0.15%)

Total flood-related pixels (class > 1): 139,164
 Percentage: 8.23%

```
In [17]: # Try Loading different tiles to find one with mixed flood/non-flood patches
print("Testing different tiles to find varied flood distribution:")
print("="*60)

for idx in range(min(5, len(germany_loader.get_tile_list()))):
    tile_name = germany_loader.get_tile_list()[idx]
    tile_data = load_tile_data(GERMANY_TRAIN, tile_name, 'Germany')

    # Calculate flood percentage
    flood_px = np.sum((tile_data['mask'] == 2) | (tile_data['mask'] == 3) | (tile_data['mask'] == 4))
    total_px = tile_data['mask'].size
    flood_pct = (flood_px / total_px) * 100

    print(f"\nTile {idx}: {tile_name}")
    print(f"  Flood pixels: {flood_px:,} ({flood_pct:.2f}%)")
    print(f"  Classes present: {np.unique(tile_data['mask'])}")

    # If this tile has moderate flooding (2-15%), use it for demo
    if 2.0 <= flood_pct <= 15.0:
```

```
        print(f" Good candidate for mixed flood/non-flood patches")
        break
```

Testing different tiles to find varied flood distribution:

```
=====
```

Tile 0: 0_41_59.geojson
Flood pixels: 117,243 (6.94%)

Tile 0: 0_41_59.geojson
Flood pixels: 117,243 (6.94%)
Classes present: [0 1 2 5 6]
Good candidate for mixed flood/non-flood patches
Classes present: [0 1 2 5 6]
Good candidate for mixed flood/non-flood patches

```
In [18]: print("\n" + "*60)
print("Testing patch extraction with different thresholds:")
print("*60)

for threshold in [100, 2621, 5000, 10000, 20000]:
    test_extractor = PatchExtractor(
        patch_size=PATCH_SIZE,
        overlap=PATCH_OVERLAP,
        min_flood_pixels=threshold
    )

    test_patches = test_extractor.extract_patches(
        combined_image,
        mask=sample_data['mask'],
        oversample_flood=False # Disable oversampling for clarity
    )

    flood_count = sum(1 for p in test_patches if p['is_flood_positive'])
    non_flood_count = len(test_patches) - flood_count

    print(f"\nThreshold: {threshold} pixels ({(threshold/(PATCH_SIZE**2))*100:.2f}% of patch)")
    print(f" Total patches: {len(test_patches)}")
    print(f" Flood-positive: {flood_count}")
    print(f" Non-flood: {non_flood_count}")
    print(f" Flood ratio: {((flood_count/len(test_patches))*100:.1f}%)")
```

```
=====
Testing patch extraction with different thresholds:
=====

Threshold: 100 pixels (0.04% of patch)
    Total patches: 9
    Flood-positive: 9
    Non-flood: 0
    Flood ratio: 100.0%

Threshold: 2621 pixels (1.00% of patch)
    Total patches: 9
    Flood-positive: 7
    Non-flood: 2
    Flood ratio: 77.8%

Threshold: 5000 pixels (1.91% of patch)
    Total patches: 9
    Flood-positive: 7
    Non-flood: 2
    Flood ratio: 77.8%

Threshold: 5000 pixels (1.91% of patch)
    Total patches: 9
    Flood-positive: 7
    Non-flood: 2
    Flood ratio: 77.8%

Threshold: 10000 pixels (3.81% of patch)
    Total patches: 9
    Flood-positive: 7
    Non-flood: 2
    Flood ratio: 77.8%

Threshold: 20000 pixels (7.63% of patch)
    Total patches: 9
    Flood-positive: 3
    Non-flood: 6
    Flood ratio: 33.3%

Threshold: 10000 pixels (3.81% of patch)
    Total patches: 9
```

```
Flood-positive: 7
Non-flood: 2
Flood ratio: 77.8%
```

```
Threshold: 20000 pixels (7.63% of patch)
Total patches: 9
Flood-positive: 3
Non-flood: 6
Flood ratio: 33.3%
```

```
In [19]: print("\nPatch-level flood analysis:")
for i, patch in enumerate(patches):
    flood_px = patch['flood_pixels']
    total_px = PATCH_SIZE * PATCH_SIZE
    flood_pct = (flood_px / total_px) * 100
    print(f" Patch {i}: {flood_px} flood pixels ({flood_pct:.2f}%)")

print(f"\nPatch size: {PATCH_SIZE}x{PATCH_SIZE} = {PATCH_SIZE*PATCH_SIZE:,} pixels")
print(f"Min flood pixels threshold: {patch_extractor.min_flood_pixels}")
print(f"Min flood percentage needed: {((patch_extractor.min_flood_pixels / (PATCH_SIZE*PATCH_SIZE)) * 100:.2f}%)")
```

```
Patch-level flood analysis:
Patch 0: 64040 flood pixels (24.43%)
Patch 1: 29461 flood pixels (11.24%)
Patch 2: 16325 flood pixels (6.23%)
Patch 3: 21596 flood pixels (8.24%)
Patch 4: 12637 flood pixels (4.82%)
Patch 5: 1140 flood pixels (0.43%)
Patch 6: 548 flood pixels (0.21%)
Patch 7: 11187 flood pixels (4.27%)
Patch 8: 14272 flood pixels (5.44%)
```

```
Patch size: 512x512 = 262,144 pixels
Min flood pixels threshold: 2621
Min flood percentage needed: 1.00%
```

```
In [20]: # Visualize sample patches
n_samples = min(8, len(patches))
sample_patches = np.random.choice(patches, n_samples, replace=False)

fig, axes = plt.subplots(4, n_samples, figsize=(n_samples*3, 12))
```

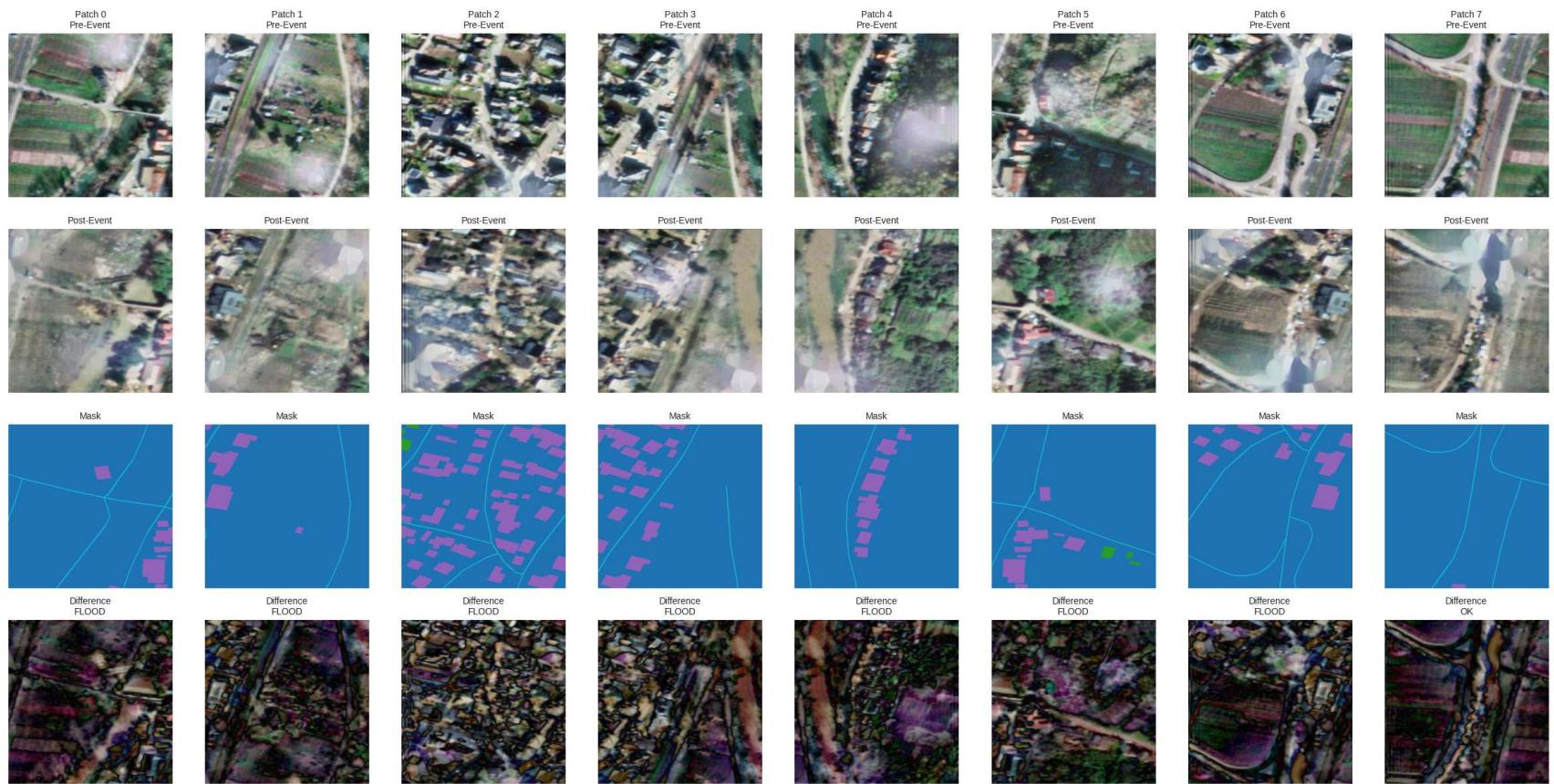
```
for i, patch in enumerate(sample_patches):
    # Pre-event (first 3 channels)
    pre_patch = patch['image'][:, :, :3]
    axes[0, i].imshow(pre_patch)
    axes[0, i].set_title(f"Patch {i}\nPre-Event", fontsize=10)
    axes[0, i].axis('off')

    # Post-event (last 3 channels)
    post_patch = patch['image'][:, :, 3:6]
    axes[1, i].imshow(post_patch)
    axes[1, i].set_title('Post-Event', fontsize=10)
    axes[1, i].axis('off')

    # Mask
    axes[2, i].imshow(patch['mask'], cmap='tab10', vmin=0, vmax=5)
    axes[2, i].set_title('Mask', fontsize=10)
    axes[2, i].axis('off')

    # Difference
    diff_patch = np.abs(post_patch - pre_patch)
    axes[3, i].imshow(diff_patch)
    flood_status = 'FLOOD' if patch['is_flood_positive'] else 'OK'
    axes[3, i].set_title(f"Difference\n{n{flood_status}}", fontsize=10)
    axes[3, i].axis('off')

plt.tight_layout()
plt.show()
```



7. Class Distribution Analysis

```
In [21]: # Analyze class distribution across all patches
class_totals = {i: 0 for i in range(7)} # Classes 0-6

for patch in patches:
    for cls, count in patch.get('class_distribution', {}).items():
        class_totals[int(cls)] += count

# Create visualization
fig, axes = plt.subplots(1, 2, figsize=(16, 6))

# Pie chart
classes = list(class_totals.keys())
```

```

counts = list(class_totals.values())
labels = [f'{CLASS_NAMES.get(c, f'Class {c}')}\n{counts[i]}:' for i, c in enumerate(classes)]

# Convert RGB colors (0-255) to hex format, normalizing to 0-1 range
colors = []
for cls in classes:
    rgb = CLASS_COLORS.get(cls, [128, 128, 128])
    # Normalize RGB values from 0-255 to 0-1
    r, g, b = [val/255.0 for val in rgb]
    colors.append((r, g, b))

axes[0].pie(counts, labels=labels, colors=colors, autopct='%1.1f%%', startangle=90)
axes[0].set_title('Class Distribution (Pixel Count)', fontsize=14, fontweight='bold')

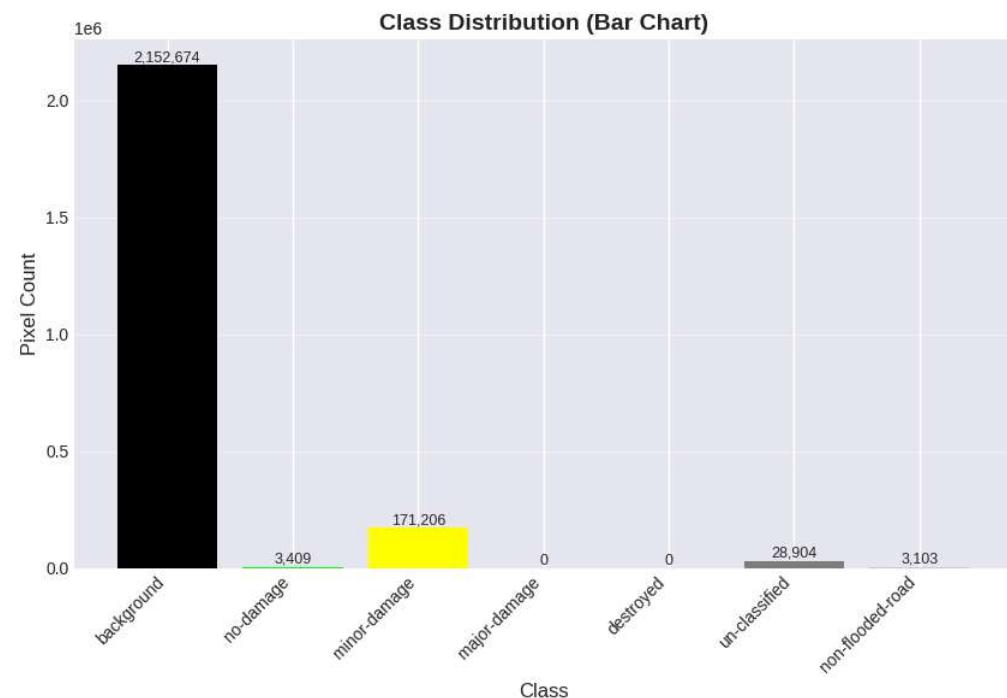
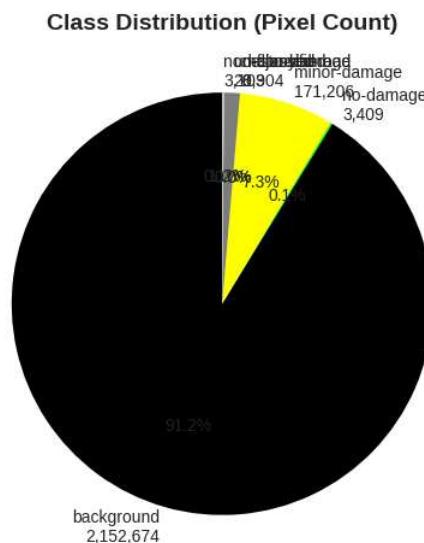
# Bar chart
axes[1].bar(classes, counts, color=colors)
axes[1].set_xlabel('Class', fontsize=12)
axes[1].set_ylabel('Pixel Count', fontsize=12)
axes[1].set_title('Class Distribution (Bar Chart)', fontsize=14, fontweight='bold')
axes[1].set_xticks(classes)
axes[1].set_xticklabels([CLASS_NAMES.get(c, f'C{c}') for c in classes], rotation=45, ha='right')
axes[1].grid(alpha=0.3, axis='y')

# Add counts on bars
for i, (cls, count) in enumerate(zip(classes, counts)):
    axes[1].text(cls, count, f'{count}:' , ha='center', va='bottom', fontsize=9)

plt.tight_layout()
plt.show()

print("\nClass Imbalance Summary:")
total_pixels = sum(counts)
for cls, count in zip(classes, counts):
    pct = (count / total_pixels) * 100
    print(f" {CLASS_NAMES.get(cls, f'Class {cls}')}: {count} pixels ({pct:.2f}%)")

```



Class Imbalance Summary:

background: 2,152,674 pixels (91.24%)
 no-damage: 3,409 pixels (0.14%)
 minor-damage: 171,206 pixels (7.26%)
 major-damage: 0 pixels (0.00%)
 destroyed: 0 pixels (0.00%)
 un-classified: 28,904 pixels (1.23%)
 non-flooded-road: 3,103 pixels (0.13%)

8. Data Augmentation Preview

```
In [22]: # Get training augmentation
train_aug = get_training_augmentation(image_size=PATCH_SIZE)

# Select a flood-positive patch for demonstration
demo_patch = flood_positive[0] if len(flood_positive) > 0 else patches[0]
demo_image = demo_patch['image'][:, :, :3] # Use pre-event for demo
demo_mask = demo_patch['mask']
```

```
# Apply augmentation multiple times
n_aug = 6
fig, axes = plt.subplots(2, n_aug, figsize=(n_aug*3, 6))

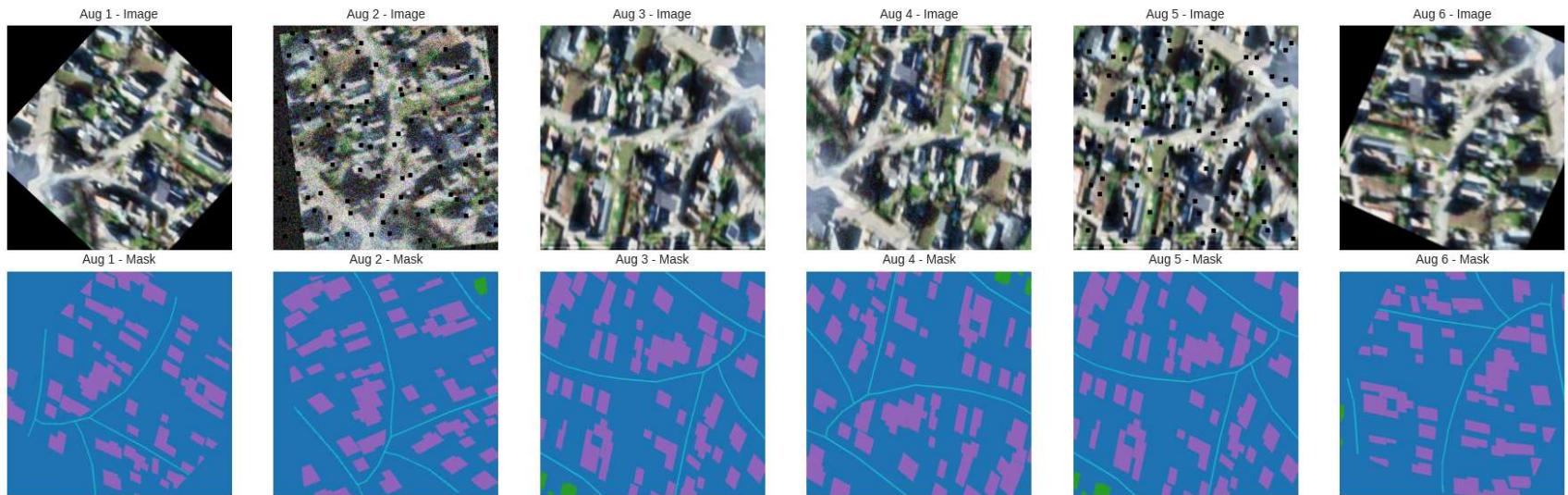
for i in range(n_aug):
    augmented = train_aug(image=demo_image, mask=demo_mask)

    axes[0, i].imshow(augmented['image'])
    axes[0, i].set_title(f'Aug {i+1} - Image', fontsize=10)
    axes[0, i].axis('off')

    axes[1, i].imshow(augmented['mask'], cmap='tab10', vmin=0, vmax=5)
    axes[1, i].set_title(f'Aug {i+1} - Mask', fontsize=10)
    axes[1, i].axis('off')

plt.suptitle('Data Augmentation Examples', fontsize=14, fontweight='bold', y=1.02)
plt.tight_layout()
plt.show()

print("\nAugmentations include:")
print(" - Horizontal/Vertical flips")
print(" - Random rotations (90 degrees)")
print(" - Shift, scale, rotate")
print(" - Brightness/contrast adjustments")
print(" - Gaussian noise and blur")
print(" - Random fog and shadows")
print(" - Grid dropout")
```

Data Augmentation Examples

Augmentations include:

- Horizontal/Vertical flips
- Random rotations (90 degrees)
- Shift, scale, rotate
- Brightness/contrast adjustments
- Gaussian noise and blur
- Random fog and shadows
- Grid dropout

9. Run Full Preprocessing Pipeline

This section runs the complete preprocessing pipeline on both regions.

```
In [23]: # Import preprocessing script
import subprocess

print("Starting full preprocessing pipeline...")
print("This may take 30-60 minutes depending on dataset size.")
print("\nProcessing:")
print(" 1. Load all tiles from Germany and Louisiana-East")
print(" 2. Apply quality checks")
print(" 3. Apply CLAHE enhancement")
```

```
print(" 4. Extract patches with smart sampling")
print(" 5. Oversample flood-positive patches")
print(" 6. Create geo-stratified train/val/test splits")
print(" 7. Export to dataset/processed/")
print("\n" + "*80)
```

Starting full preprocessing pipeline...

This may take 30-60 minutes depending on dataset size.

Processing:

1. Load all tiles from Germany and Louisiana-East
 2. Apply quality checks
 3. Apply CLAHE enhancement
 4. Extract patches with smart sampling
 5. Oversample flood-positive patches
 6. Create geo-stratified train/val/test splits
 7. Export to dataset/processed/
-

```
In [24]: # Execute the preprocessing script
# This will now:
# 1. Process training data (Germany + Louisiana-East)
# 2. Create train/val split (85%/15%)
# 3. Process test data (Louisiana-West_Test_Public)

if IS_COLAB:
    %run src/run_preprocessing.py
else:
    %run ../src/run_preprocessing.py
```

```
=====
FLOOD DETECTION - DATA PREPROCESSING PIPELINE
=====
```

```
=====
STEP 1: PROCESSING TRAINING DATA
=====
```

Discovered 2 training region(s):

- Germany_Training_Public: /content/aaai521_3proj/dataset/raw/train/Germany_Training_Public
- Louisiana-East_Training_Public: /content/aaai521_3proj/dataset/raw/train/Louisiana-East_Training_Public

```
=====
Processing region: Germany_Training_Public
=====
```

Found 202 tiles

Using 12 parallel workers

Flood statistics:

```
total_segments: 9761
flooded_count: 2498
non_flooded_count: 7183
null_count: 80
flooded_pct: 25.591640200799098
non_flooded_pct: 73.58877164224977
null_pct: 0.8195881569511321
flooded_road_length_km: 34.370402867674
total_road_length_km: 163.28881066947514
```

```
Processing Germany_Training_Public: 100%|██████████| 202/202 [00:58<00:00,  3.47tile/s, progress=100.0%, success=197,
failed=5]
```

```
=====
Processing region: Louisiana-East_Training_Public
=====
Found 599 tiles
Using 12 parallel workers
```

Flood statistics:

```
total_segments: 23663
flooded_count: 4577
non_flooded_count: 18894
null_count: 192
flooded_pct: 19.342433334742
non_flooded_pct: 79.84617335080083
null_pct: 0.8113933144571693
flooded_road_length_km: 116.24277245596251
total_road_length_km: 605.1457362223111
```

```
Processing Louisiana-East_Training_Public: 100%|██████████| 599/599 [03:02<00:00, 3.28tile/s, progress=100.0%, successes=773, failed=28]
```

```
=====
TRAINING DATA PROCESSING SUMMARY
=====
```

```
Total tiles processed: 773
Total tiles failed: 28
Quality check failures: 28
Total patches extracted: 7285
Flood-positive patches: 732
Flood ratio: 10.05%
```

```
=====
STEP 2: CREATING TRAIN/VAL SPLITS
=====
```

Geo-stratified split:

```
Training: 657 tiles, 6207 patches
Validation: 115 tiles, 1069 patches
Test: 1 tiles, 9 patches
```

Flood-positive patches:

```
Training: 635/6207 (10.2%)
Validation: 97/1069 (9.1%)
Test: 0/9 (0.0%)
```

Copying validation files...

```
Copying validation patches: 100%|██████████| 1069/1069 [00:50<00:00, 21.11it/s]
```

```
Copying validation patches: 100%|██████████| 1069/1069 [00:50<00:00, 21.11it/s]
```

Copying validation full-resolution images...

Copying 115 unique tiles...

```
Copying full-res images: 100%|██████████| 115/115 [00:08<00:00, 14.29it/s]
```

Saving train/val metadata...

Metadata saved:

JSON: /content/aai521_3proj/dataset/processed/train/metadata/train_metadata.json
Pickle: /content/aai521_3proj/dataset/processed/train/metadata/train_metadata.pkl
CSV: /content/aai521_3proj/dataset/processed/train/metadata/train_metadata.csv

Metadata saved:

JSON: /content/aai521_3proj/dataset/processed/val/metadata/val_metadata.json
Pickle: /content/aai521_3proj/dataset/processed/val/metadata/val_metadata.pkl
CSV: /content/aai521_3proj/dataset/processed/val/metadata/val_metadata.csv

=====

STEP 3: CALCULATING DATASET STATISTICS

=====

Calculating statistics from 100 sample images...

Metadata saved:

JSON: /content/aai521_3proj/dataset/processed/train/metadata/train_metadata.json
Pickle: /content/aai521_3proj/dataset/processed/train/metadata/train_metadata.pkl
CSV: /content/aai521_3proj/dataset/processed/train/metadata/train_metadata.csv

Metadata saved:

JSON: /content/aai521_3proj/dataset/processed/val/metadata/val_metadata.json
Pickle: /content/aai521_3proj/dataset/processed/val/metadata/val_metadata.pkl
CSV: /content/aai521_3proj/dataset/processed/val/metadata/val_metadata.csv

=====

STEP 3: CALCULATING DATASET STATISTICS

=====

Calculating statistics from 100 sample images...

Loading samples: 100%|██████████| 100/100 [00:03<00:00, 27.98it/s]

Dataset statistics:

Mean (per channel): [0.45236682891845703, 0.4747048318386078, 0.4516935348510742, 0.413782000541687, 0.4532981514930725, 0.39436012506484985]

Std (per channel): [0.2668497860431671, 0.2553929388523102, 0.2546478509902954, 0.22675950825214386, 0.22327350080013275, 0.22437310218811035]

Min: [0.0, 0.0, 0.0, 0.0, 0.0, 0.0]

Max: [1.0, 1.0, 1.0, 1.0, 1.0, 1.0]

Statistics saved to: /content/aai521_3proj/dataset/processed/dataset_statistics.json

=====

Processed data saved to:

Training: /content/aai521_3proj/dataset/processed/train

- Patches: 6207

- Flood-positive: 635

Validation: /content/aai521_3proj/dataset/processed/val

- Patches: 1069

- Flood-positive: 97

Output structure:

- images/ : Extracted patches (512x512, 6 channels)

- masks/ : Segmentation masks for patches

- processed_images/: Full-resolution processed images

- Germany_Training_Public/

- PRE-event/ : Processed pre-event images

- POST-event/ : Processed post-event images

- Louisiana-East_Training_Public/

- PRE-event/ : Processed pre-event images

- POST-event/ : Processed post-event images

- metadata/ : JSON, pickle, and CSV metadata files

<Figure size 640x480 with 0 Axes>

```
In [25]: print("=*80")
print("\nProcessing test data...")

if IS_COLAB:
    %run src/process_test_data.py
else:
    %run ../src/process_test_data.py
```

```
=====
Processing test data...
=====
```

```
=====
PHASE III: PROCESSING TEST DATA (Louisiana-West)
=====
```

```
Discovered 1 test region(s):
- Louisiana-West_Test_Public: /content/aai521_3proj/dataset/raw/test/Louisiana-West_Test_Public
```

```
=====
Processing region: Louisiana-West_Test_Public
=====
```

```
Found 406 tiles
```

```
Using 12 parallel workers
```

```
Flood statistics:
```

```
Processing Louisiana-West_Test_Public: 100%|██████████| 406/406 [02:17<00:00,  2.95tile/s, progress=100.0%, success=397, failed=9]
```

```
=====
TEST DATA PROCESSING SUMMARY
=====
```

```
Total tiles processed: 397
Total tiles failed: 9
Quality check failures: 9
Total patches extracted: 3573
Flood-positive patches: 0
Flood ratio: 0.00%
```

```
Saving test metadata...
```

```
Metadata saved:
```

```
JSON: /content/aai521_3proj/dataset/processed/test/metadata/test_metadata.json
Pickle: /content/aai521_3proj/dataset/processed/test/metadata/test_metadata.pkl
CSV: /content/aai521_3proj/dataset/processed/test/metadata/test_metadata.csv
```

```
=====
PHASE III COMPLETE!
=====
```

```
Processed test data saved to: /content/aai521_3proj/dataset/processed/test
```

- Patches: 3573
- Flood-positive: 0

```
Metadata saved:
```

```
JSON: /content/aai521_3proj/dataset/processed/test/metadata/test_metadata.json
Pickle: /content/aai521_3proj/dataset/processed/test/metadata/test_metadata.pkl
CSV: /content/aai521_3proj/dataset/processed/test/metadata/test_metadata.csv
```

```
=====
PHASE III COMPLETE!
=====
```

```
Processed test data saved to: /content/aai521_3proj/dataset/processed/test
```

- Patches: 3573
- Flood-positive: 0

10. Verify Preprocessing

```
In [26]: import os
from pathlib import Path

print("*"*80)
print("PREPROCESSING VERIFICATION")
print("*"*80)

# Check 1: Val folder has full-resolution processed images
print("\n1. Checking Val folder processed_images...")
val_processed = Path('../dataset/processed/val/processed_images')
if val_processed.exists():
    regions = [d.name for d in val_processed.iterdir() if d.is_dir()]
    print(f"  Found {len(regions)} region(s): {regions}")

    for region in regions:
        pre_dir = val_processed / region / 'PRE-event'
        post_dir = val_processed / region / 'POST-event'

        if pre_dir.exists() and post_dir.exists():
            pre_count = len(list(pre_dir.glob('*tif')))
            post_count = len(list(post_dir.glob('*tif')))
            print(f"  {region}:")
            print(f"    - PRE-event: {pre_count} images")
            print(f"    - POST-event: {post_count} images")

            if pre_count > 0:
                # Show sample filenames
                sample = list(pre_dir.glob('*tif'))[0].name
                print(f"      - Sample: {sample}")
            else:
                print(f"      {region}: Missing PRE/POST directories")
        else:
            print(f"  {region}: Val processed_images directory not found!")

# Check 2: Test folder has Louisiana-West data
print("\n2. Checking Test folder for Louisiana-West data...")
test_processed = Path('../dataset/processed/test/processed_images')
if test_processed.exists():
    regions = [d.name for d in test_processed.iterdir() if d.is_dir()]
    print(f"  Found {len(regions)} region(s): {regions}")
```

```

for region in regions:
    pre_dir = test_processed / region / 'PRE-event'
    post_dir = test_processed / region / 'POST-event'

    if pre_dir.exists() and post_dir.exists():
        pre_count = len(list(pre_dir.glob('*.*tif')))
        post_count = len(list(post_dir.glob('*.*tif')))
        print(f"  {region}:")
        print(f"    - PRE-event: {pre_count} images")
        print(f"    - POST-event: {post_count} images")

        if pre_count > 0:
            # Show sample filenames
            sample = list(pre_dir.glob('*.*tif'))[0].name
            print(f"      - Sample: {sample}")

            # Verify it's Louisiana-West data (different sensor IDs)
            if 'Louisiana-West' in region:
                print(f"      Correctly using Louisiana-West test data!")
            elif 'Germany' in region or 'Louisiana-East' in region:
                print(f"      WARNING: Test should not contain training regions!")
            else:
                print(f"      {region}: Missing PRE/POST directories")
        else:
            print("  Test processed_images directory not found!")

# Check 3: Verify filenames match CSV
print("\n3. Verifying filenames match CSV mappings...")
import pandas as pd

# Check Germany training
csv_path = Path('../dataset/raw/train/Germany_Training_Public/Germany_Training_Public_label_image_mapping.csv')
if csv_path.exists():
    df = pd.read_csv(csv_path)
    sample_pre = df.iloc[0]['pre-event image']
    sample_post = df.iloc[0]['post-event image 1']

    # Check if processed file exists with same name
    processed_pre = Path('../dataset/processed/train/processed_images/Germany_Training_Public/PRE-event') / sample_pre
    processed_post = Path('../dataset/processed/train/processed_images/Germany_Training_Public/POST-event') / sample_post

    print(f"  Germany sample from CSV:")

```

```

print(f"      - PRE: {sample_pre}")
print(f"      - POST: {sample_post}")

if processed_pre.exists():
    print(f"      Pre-event file found with exact name!")
else:
    print(f"      Pre-event file NOT found!")

if processed_post.exists():
    print(f"      Post-event file found with exact name!")
else:
    print(f"      Post-event file NOT found!")

# Check Louisiana-West test
csv_path_test = Path('../dataset/raw/test/Louisiana-West_Test_Public/Louisiana-West_Test_Public_label_image_mapping.csv')
if csv_path_test.exists():
    df_test = pd.read_csv(csv_path_test)
    sample_pre_test = df_test.iloc[0]['pre-event image']
    sample_post_test = df_test.iloc[0]['post-event image 1']

# Check if processed file exists with same name
processed_pre_test = Path('../dataset/processed/test/processed_images/Louisiana-West_Test_Public/PRE-event') / sample_pre_test
processed_post_test = Path('../dataset/processed/test/processed_images/Louisiana-West_Test_Public/POST-event') / sample_post_test

print(f"\n  Louisiana-West sample from CSV:")
print(f"      - PRE: {sample_pre_test}")
print(f"      - POST: {sample_post_test}")

if processed_pre_test.exists():
    print(f"      Pre-event file found with exact name!")
else:
    print(f"      Pre-event file NOT found!")

if processed_post_test.exists():
    print(f"      Post-event file found with exact name!")
else:
    print(f"      Post-event file NOT found!")

print("\n" + "*80)
print("VERIFICATION COMPLETE")
print("*80)

```

=====
PREPROCESSING VERIFICATION
=====

1. Checking Val folder processed_images...
Val processed_images directory not found!
2. Checking Test folder for Louisiana-West data...
Test processed_images directory not found!
3. Verifying filenames match CSV mappings...

=====
VERIFICATION COMPLETE
=====

11. Export Metadata Files

After preprocessing is complete, generate metadata files (JSON, Pickle, CSV) for all splits.

In [27]:

```
# Run metadata export script
if IS_COLAB:
    %run src/export_metadata.py
else:
    %run ../src/export_metadata.py
```

=====
METADATA EXPORT FOR PREPROCESSED DATASETS
==========
Processing TRAIN split
=====

Scanning directory: /content/aai521_3proj/dataset/processed/train

Found 7285 image files
Found 7285 image files

Extracting metadata: 100%|██████████| 7285/7285 [04:31<00:00, 26.81it/s]

```
Saving train metadata (7285 patches)...
  JSON: /content/aaiai521_3proj/dataset/processed/train/metadata/train_metadata.json
  Pickle: /content/aaiai521_3proj/dataset/processed/train/metadata/train_metadata.pkl
  CSV: /content/aaiai521_3proj/dataset/processed/train/metadata/train_metadata.csv
Statistics:
  Total patches: 7285
  Flood-positive: 2713 (37.2%)
  Class distribution:
    background: 1,860,991,567 pixels (97.45%)
    no-damage: 31,784,864 pixels (1.66%)
    minor-damage: 8,851,035 pixels (0.46%)
    un-classified: 1,849,705 pixels (0.10%)
    non-flooded-road: 6,241,869 pixels (0.33%)
```

```
=====
Processing VAL split
=====
```

```
Scanning directory: /content/aaiai521_3proj/dataset/processed/val
  Found 1741 image files
  JSON: /content/aaiai521_3proj/dataset/processed/train/metadata/train_metadata.json
  Pickle: /content/aaiai521_3proj/dataset/processed/train/metadata/train_metadata.pkl
  CSV: /content/aaiai521_3proj/dataset/processed/train/metadata/train_metadata.csv
Statistics:
  Total patches: 7285
  Flood-positive: 2713 (37.2%)
  Class distribution:
    background: 1,860,991,567 pixels (97.45%)
    no-damage: 31,784,864 pixels (1.66%)
    minor-damage: 8,851,035 pixels (0.46%)
    un-classified: 1,849,705 pixels (0.10%)
    non-flooded-road: 6,241,869 pixels (0.33%)
```

```
=====
Processing VAL split
=====
```

```
Scanning directory: /content/aaiai521_3proj/dataset/processed/val
  Found 1741 image files
Extracting metadata: 100%|██████████| 1741/1741 [01:00<00:00, 28.65it/s]
```

```
Saving val metadata (1741 patches)...
  JSON: /content/aai521_3proj/dataset/processed/val/metadata/val_metadata.json
  Pickle: /content/aai521_3proj/dataset/processed/val/metadata/val_metadata.pkl
  CSV: /content/aai521_3proj/dataset/processed/val/metadata/val_metadata.csv
  Statistics:
    Total patches: 1741
    Flood-positive: 609 (35.0%)
    Class distribution:
      background: 444,566,688 pixels (97.41%)
      no-damage: 8,002,745 pixels (1.75%)
      minor-damage: 1,934,982 pixels (0.42%)
      un-classified: 385,845 pixels (0.08%)
      non-flooded-road: 1,502,444 pixels (0.33%)
```

```
=====
Processing TEST split
=====
```

```
Scanning directory: /content/aai521_3proj/dataset/processed/test
  Found 3573 image files
```

```
Extracting metadata: 100%|██████████| 3573/3573 [03:19<00:00, 17.89it/s]
Extracting metadata: 100%|██████████| 3573/3573 [03:19<00:00, 17.89it/s]
```

```
Saving test metadata (3573 patches)...
  JSON: /content/aai521_3proj/dataset/processed/test/metadata/test_metadata.json
  Pickle: /content/aai521_3proj/dataset/processed/test/metadata/test_metadata.pkl
  CSV: /content/aai521_3proj/dataset/processed/test/metadata/test_metadata.csv
  Statistics:
    Total patches: 3573
    Flood-positive: 0 (0.0%)
    Class distribution:
      background: 936,640,512 pixels (100.00%)
```

```
=====
METADATA EXPORT COMPLETE!
=====
```

```
Metadata files generated:
  - JSON (human-readable)
  - Pickle (fast loading)
  - CSV (spreadsheet-compatible)
```

```
Ready for model training!
```

12. Validate Processed Data

After preprocessing and metadata export are complete, validate the output.

```
In [28]: # Check if processed data exists
if PROCESSED_TRAIN_DIR.exists():
    print("Processed data directory exists")

    # Count files
train_images = list((PROCESSED_TRAIN_DIR / 'images').glob('*.*'))
train_masks = list((PROCESSED_TRAIN_DIR / 'masks').glob('*.*'))

print(f"\nTraining set:")
print(f"  Images: {len(train_images)}")
print(f"  Masks: {len(train_masks)}")

# Load metadata
metadata_path = PROCESSED_TRAIN_DIR / 'metadata' / 'train_metadata.json'
if metadata_path.exists():
    import json
    with open(metadata_path, 'r') as f:
        metadata = json.load(f)

    print(f"  Metadata entries: {len(metadata)}")

    # Count flood-positive
flood_count = sum(1 for m in metadata if m['is_flood_positive'])
print(f"  Flood-positive patches: {flood_count} ({flood_count/len(metadata)*100:.1f}%)")


# Load and display a sample
if len(train_images) > 0:
    sample_img = np.load(train_images[0])
    sample_mask = np.load(train_masks[0])

    print(f"\nSample patch:")
    print(f"  Image shape: {sample_img.shape}")
    print(f"  Mask shape: {sample_mask.shape}")
    print(f"  Image range: [{sample_img.min():.3f}, {sample_img.max():.3f}]")
    print(f"  Mask classes: {np.unique(sample_mask)}")
```

```
# Visualize
fig, axes = plt.subplots(1, 3, figsize=(15, 5))

axes[0].imshow(sample_img[:, :, :3]) # Pre-event
axes[0].set_title('Pre-Event (Processed)', fontsize=12)
axes[0].axis('off')

axes[1].imshow(sample_img[:, :, 3:6]) # Post-event
axes[1].set_title('Post-Event (Processed)', fontsize=12)
axes[1].axis('off')

axes[2].imshow(sample_mask, cmap='tab10')
axes[2].set_title('Mask', fontsize=12)
axes[2].axis('off')

plt.tight_layout()
plt.show()

else:
    print("Processed data not found. Run preprocessing first.")
```

Processed data directory exists

Training set:

Images: 7285

Masks: 7285

Metadata entries: 7285

Flood-positive patches: 2713 (37.2%)

Sample patch:

Image shape: (512, 512, 6)

Mask shape: (512, 512)

Image range: [0.000, 0.996]

Mask classes: [0]

Training set:

Images: 7285

Masks: 7285

Metadata entries: 7285

Flood-positive patches: 2713 (37.2%)

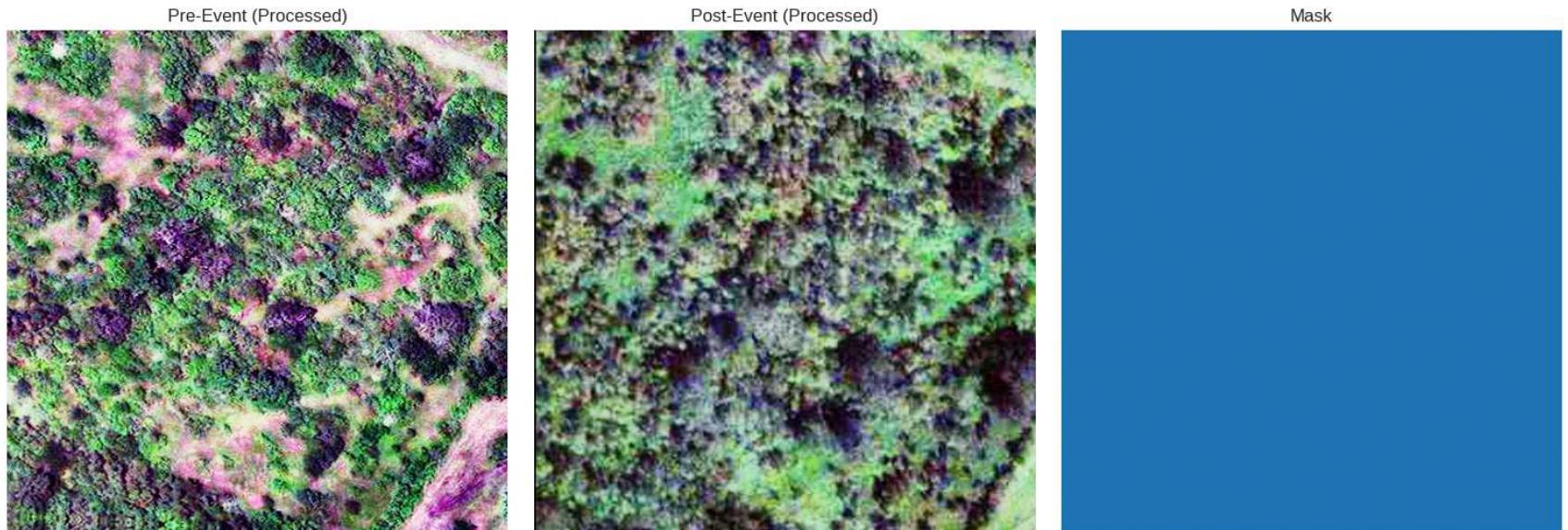
Sample patch:

Image shape: (512, 512, 6)

Mask shape: (512, 512)

Image range: [0.000, 0.996]

Mask classes: [0]



13. Compare Raw vs Processed Full-Resolution Images

Compare original raw images with processed full-resolution TIF images for both Germany and Louisiana datasets.

```
In [29]: import random
from pathlib import Path
import pandas as pd

# Check if processed full-resolution images exist
if IS_COLAB:
    processed_base = Path('dataset/processed')
else:
    processed_base = Path('../dataset/processed')
train_processed_images = processed_base / 'train' / 'processed_images'

if not train_processed_images.exists():
    print("Processed full-resolution images not found.")
    print("Run preprocessing pipeline first to generate processed images.")
else:
    print("Processed images directory found")

# Get available regions
```

```
available_regions = [d.name for d in train_processed_images.iterdir() if d.is_dir()]
print(f"Available regions: {available_regions}")

# Map to raw data directories (use actual directory names)
region_mapping = {
    'Germany_Training_Public': GERMANY_TRAIN,
    'Louisiana-East_Training_Public': LOUISIANA_EAST_TRAIN
}

# Select 2 random tiles from each region
comparison_samples = []

for region in available_regions:
    if region in region_mapping:
        raw_dir = region_mapping[region]

        # Load CSV mapping file
        csv_name = f"{region}_label_image_mapping.csv"
        csv_path = raw_dir / csv_name

        if not csv_path.exists():
            print(f"CSV mapping not found: {csv_path}")
            continue

        # Read CSV to get pre/post image mappings
        mapping_df = pd.read_csv(csv_path)
        print(f"\nLoaded {len(mapping_df)} mappings from {csv_name}")

        # Get list of processed PRE-event images
        pre_processed_dir = train_processed_images / region / 'PRE-event'
        if not pre_processed_dir.exists():
            print(f"No PRE-event processed images for {region}")
            continue

        processed_pre_files = list(pre_processed_dir.glob('*.*tif'))

        if len(processed_pre_files) == 0:
            print(f"No TIF files found for {region}")
            continue

        # Select 2 random samples
        n_samples = min(2, len(processed_pre_files))
```

```

selected_files = random.sample(processed_pre_files, n_samples)

for pre_tif in selected_files:
    # Processed files are named after the pre-event image
    pre_image_name = pre_tif.name # e.g., "10500500C4DD7000_0_41_59.tif"

    # Find matching row in CSV
    matching_row = mapping_df[mapping_df['pre-event image'] == pre_image_name]

    if matching_row.empty:
        print(f"No CSV mapping found for: {pre_image_name}")
        continue

    # Get post-event image name from CSV
    post_image_name = matching_row.iloc[0]['post-event image 1']

    # Paths
    raw_pre_path = raw_dir / 'PRE-event' / pre_image_name
    raw_post_path = raw_dir / 'POST-event' / post_image_name
    # Processed POST-event images are saved with their original POST-event filenames
    processed_post_path = train_processed_images / region / 'POST-event' / post_image_name

    if all([p.exists() for p in [raw_pre_path, raw_post_path, processed_post_path]]):
        comparison_samples.append({
            'region': region,
            'tile': pre_tif.stem,
            'raw_pre': raw_pre_path,
            'raw_post': raw_post_path,
            'processed_pre': pre_tif,
            'processed_post': processed_post_path
        })
        print(f"Found complete set: {pre_tif.stem}")
    else:
        missing = []
        if not raw_pre_path.exists(): missing.append("raw_pre")
        if not raw_post_path.exists(): missing.append("raw_post")
        if not processed_post_path.exists(): missing.append("processed_post")
        print(f"Missing files for {pre_tif.stem}: {', '.join(missing)}")

print(f"\nFound {len(comparison_samples)} complete sample sets for comparison")
for sample in comparison_samples:
    print(f" - {sample['region']}:{sample['tile']}")

```

```
Processed images directory found
Available regions: ['Louisiana-East_Training_Public', 'Germany_Training_Public']

Loaded 599 mappings from Louisiana-East_Training_Public_label_image_mapping.csv
Found complete set: 105001001A0FFC00_0_19_21
Found complete set: 10400100684A4B00_1_16_79

Loaded 202 mappings from Germany_Training_Public_label_image_mapping.csv
Found complete set: 10500500C4DD7000_0_42_69
Found complete set: 10500500C4DD7000_0_21_63

Found 4 complete sample sets for comparison
- Louisiana-East_Training_Public: 105001001A0FFC00_0_19_21
- Louisiana-East_Training_Public: 10400100684A4B00_1_16_79
- Germany_Training_Public: 10500500C4DD7000_0_42_69
- Germany_Training_Public: 10500500C4DD7000_0_21_63
```

```
In [30]: def load_tif_image(path):
    """Load TIF image and convert from uint16 to float32"""
    img = cv2.imread(str(path), cv2.IMREAD_UNCHANGED)
    if img is None:
        raise ValueError(f"Failed to load image: {path}")

    # Convert BGR to RGB
    if len(img.shape) == 3:
        img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)

    # Convert uint16 [0, 65535] to float32 [0, 1]
    img_float = img.astype(np.float32) / 65535.0
    return img_float


def load_raw_png_image(path):
    """Load raw PNG image"""
    img = cv2.imread(str(path), cv2.IMREAD_COLOR)
    if img is None:
        raise ValueError(f"Failed to load image: {path}")

    # Convert BGR to RGB
    img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)

    # Convert to float32 [0, 1]
```

```
img_float = img.astype(np.float32) / 255.0
return img_float

# Visualize all comparison samples
if len(comparison_samples) > 0:
    n_samples = len(comparison_samples)

    # Create figure with subplots: 4 columns (Raw Pre, Processed Pre, Raw Post, Processed Post) x n_samples rows
    fig, axes = plt.subplots(n_samples, 4, figsize=(20, 5*n_samples))

    # Handle single sample case (axes won't be 2D)
    if n_samples == 1:
        axes = axes.reshape(1, -1)

    for idx, sample in enumerate(comparison_samples):
        try:
            # Load images
            raw_pre = load_raw_png_image(sample['raw_pre'])
            raw_post = load_raw_png_image(sample['raw_post'])
            processed_pre = load_tif_image(sample['processed_pre'])
            processed_post = load_tif_image(sample['processed_post'])

            # Display images
            axes[idx, 0].imshow(raw_pre)
            axes[idx, 0].set_title(f"{sample['region']}\n{sample['tile']}\nRaw PRE-event",
                                  fontsize=11, fontweight='bold')
            axes[idx, 0].axis('off')

            axes[idx, 1].imshow(processed_pre)
            axes[idx, 1].set_title(f"Processed PRE-event\n(CLAEH + Cloud Removal + Deblur)",
                                  fontsize=11, fontweight='bold', color='darkgreen')
            axes[idx, 1].axis('off')

            axes[idx, 2].imshow(raw_post)
            axes[idx, 2].set_title(f"Raw POST-event",
                                  fontsize=11, fontweight='bold')
            axes[idx, 2].axis('off')

            axes[idx, 3].imshow(processed_post)
            axes[idx, 3].set_title(f"Processed POST-event\n(CLAEH + Cloud Removal + Deblur)",
                                  fontsize=11, fontweight='bold', color='darkgreen')

        except Exception as e:
            print(f"Error processing sample {idx}: {e}")

    plt.tight_layout()
    plt.show()

# Save the processed images
for sample in comparison_samples:
    sample['processed_pre'].save(f"processed_{sample['region']}_{sample['tile']}_{sample['raw_file'].split('.')[0]}.tif")
    sample['processed_post'].save(f"processed_{sample['region']}_{sample['tile']}_{sample['raw_file'].split('.')[0]}_post.tif")
```

```

        axes[idx, 3].axis('off')

        # Calculate quality improvements
        pre_raw_metrics = calculate_quality_metrics(raw_pre)
        pre_proc_metrics = calculate_quality_metrics(processed_pre)
        post_raw_metrics = calculate_quality_metrics(raw_post)
        post_proc_metrics = calculate_quality_metrics(processed_post)

        print(f"\n{sample['region']} - {sample['tile']}:")

        print(f"  PRE-event improvements:")
        print(f"    Sharpness: {pre_raw_metrics['sharpness']:.1f} → {pre_proc_metrics['sharpness']:.1f} "
              f"({((pre_proc_metrics['sharpness'])/pre_raw_metrics['sharpness'])-1)*100:+.1f}%)")
        print(f"    Contrast: {pre_raw_metrics['contrast']:.3f} → {pre_proc_metrics['contrast']:.3f} "
              f"({((pre_proc_metrics['contrast'])/pre_raw_metrics['contrast'])-1)*100:+.1f}%)")

        print(f"  POST-event improvements:")
        print(f"    Sharpness: {post_raw_metrics['sharpness']:.1f} → {post_proc_metrics['sharpness']:.1f} "
              f"({((post_proc_metrics['sharpness'])/post_raw_metrics['sharpness'])-1)*100:+.1f}%)")
        print(f"    Contrast: {post_raw_metrics['contrast']:.3f} → {post_proc_metrics['contrast']:.3f} "
              f"({((post_proc_metrics['contrast'])/post_raw_metrics['contrast'])-1)*100:+.1f}%)")

    except Exception as e:
        print(f"Error loading sample {sample['tile']}: {e}")
        for col in range(4):
            axes[idx, col].text(0.5, 0.5, 'Error loading image',
                                ha='center', va='center', color='red')
        axes[idx, col].axis('off')

    plt.suptitle('Raw vs Processed Full-Resolution Images Comparison\n(Germany & Louisiana-East Datasets)',
                 fontsize=16, fontweight='bold', y=0.995)
    plt.tight_layout()
    plt.show()

print("\n" + "*80)
print("PREPROCESSING EFFECTS SUMMARY")
print("*80)
print("CLAHE Enhancement: Improved local contrast and visibility")
print("Cloud Removal: Multi-stage detection and advanced inpainting")
print("Deblurring: Wiener + Richardson-Lucy + Unsharp masking + Edge enhancement")
print("Format: Saved as TIF (uint16) for quality preservation")
print("*80)

```

```
else:  
    print("\n No comparison samples available. Run preprocessing pipeline first.")
```

Louisiana-East_Training_Public - 105001001A0FFC00_0_19_21:

PRE-event improvements:

Sharpness: 146.0 → 2529.9 (+1632.5%)

Contrast: 0.135 → 0.238 (+76.2%)

POST-event improvements:

Sharpness: 2907.6 → 7046.6 (+142.3%)

Contrast: 0.174 → 0.294 (+69.2%)

Louisiana-East_Training_Public - 10400100684A4B00_1_16_79:

PRE-event improvements:

Sharpness: 853.6 → 3038.5 (+256.0%)

Contrast: 0.198 → 0.285 (+44.0%)

POST-event improvements:

Sharpness: 277.4 → 564.5 (+103.5%)

Contrast: 0.213 → 0.235 (+10.4%)

Louisiana-East_Training_Public - 10400100684A4B00_1_16_79:

PRE-event improvements:

Sharpness: 853.6 → 3038.5 (+256.0%)

Contrast: 0.198 → 0.285 (+44.0%)

POST-event improvements:

Sharpness: 277.4 → 564.5 (+103.5%)

Contrast: 0.213 → 0.235 (+10.4%)

Germany_Training_Public - 10500500C4DD7000_0_42_69:

PRE-event improvements:

Sharpness: 81.1 → 1447.1 (+1683.5%)

Contrast: 0.143 → 0.267 (+86.7%)

POST-event improvements:

Sharpness: 75.9 → 437.3 (+476.1%)

Contrast: 0.094 → 0.216 (+129.6%)

Germany_Training_Public - 10500500C4DD7000_0_42_69:

PRE-event improvements:

Sharpness: 81.1 → 1447.1 (+1683.5%)

Contrast: 0.143 → 0.267 (+86.7%)

POST-event improvements:

Sharpness: 75.9 → 437.3 (+476.1%)

Contrast: 0.094 → 0.216 (+129.6%)

Germany_Training_Public - 10500500C4DD7000_0_21_63:

PRE-event improvements:

Sharpness: 81.4 → 1400.6 (+1620.9%)

Contrast: 0.169 → 0.277 (+64.1%)

POST-event improvements:

Sharpness: 221.4 → 5854.4 (+2544.7%)

Contrast: 0.128 → 0.273 (+113.0%)

Germany_Training_Public - 10500500C4DD7000_0_21_63:

PRE-event improvements:

Sharpness: 81.4 → 1400.6 (+1620.9%)

Contrast: 0.169 → 0.277 (+64.1%)

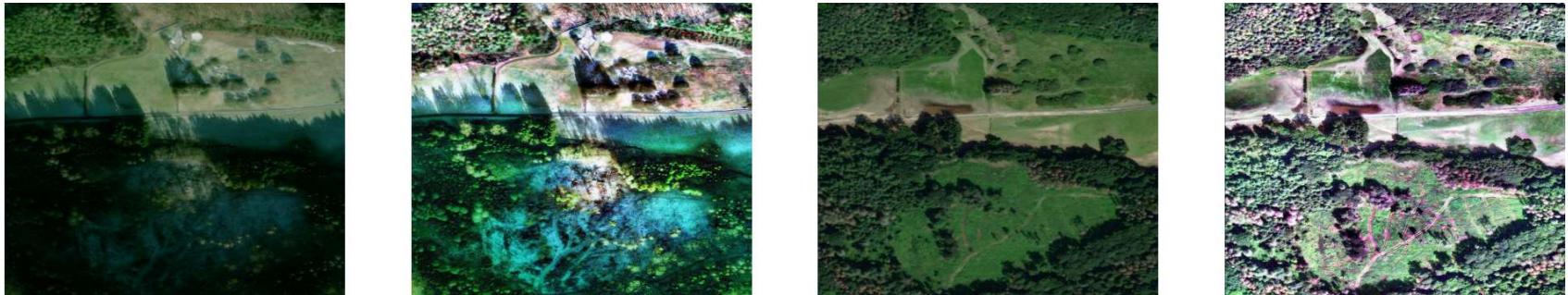
POST-event improvements:

Sharpness: 221.4 → 5854.4 (+2544.7%)

Contrast: 0.128 → 0.273 (+113.0%)

**Raw vs Processed Full-Resolution Images Comparison
(Germany & Louisiana-East Datasets)**





PREPROCESSING EFFECTS SUMMARY

CLAHE Enhancement: Improved local contrast and visibility

Cloud Removal: Multi-stage detection and advanced inpainting

Deblurring: Wiener + Richardson-Lucy + Unsharp masking + Edge enhancement

Format: Saved as TIF (uint16) for quality preservation
