Session 2: Hands-on Flood Mapping Lab

U-Net + Sentinel-1 SAR (Central Luzon)

Stylianos Kotsopoulos
EU-Philippines CoPhil Programme

Lab Overview

Duration: 2.5 hours
Type: Hands-on Coding Lab (Colab)
Goal: Train a U-Net to segment floods from Sentinel-1 SAR

You will do: - Explore SAR VV/VH patches + flood masks - Normalize, augment, and batch datasets - Implement and train U-Net - Evaluate with IoU/Dice, visualize predictions - Export masks for GIS

Prerequisites: - Session 1 (U-Net theory) - Colab GPU enabled

Resources: - Notebook: `Day3_Session2_Flood_Mapping_UNet.ipynb` - Case Study PDF: “Day 3, Session 2_ Flood Mapping with U-Net and Sentinel-1 SAR.pdf”

Case Study: Pampanga River Basin

Typhoon Ulysses (2020)

- AOI: Central Luzon (Pampanga River Basin)
- Sensor: Sentinel-1 IW GRD (VV, VH)
- Task: Binary segmentation → Flood vs Non-Flood

Why SAR for floods?

- All-weather, day/night imaging
- Flooded water → low backscatter (dark)
- VH helps separate wet soil vs open water

Data & Setup

Directory structure

```
/data/flood_mapping_dataset/
  train/{images, masks}
  val/{images, masks}
  test/{images, masks}
```

Checklist

- Mount Drive, set paths
- Confirm GPU: `tf.config.list_physical_devices('GPU')`
- Inspect sample chips and masks

Exploration (15 min)

Visualize SAR + Masks

- Show VV (dB), VH (dB)
- Overlay flood mask in blue
- Compute class imbalance (% flood)

> **Tip: Display ranges**
>
> - VV: –25 to +5 dB
> - VH: –30 to 0 dB

Preprocessing (20 min)

Normalization

- Min-max scaling for VV/VH (typical SAR dB ranges)
- Alternative: z-score per chip

Augmentation

- Flips, rotations (nadir-safe)
- Apply identical transforms to image and mask

Datasets

- Build `tf.data` pipelines (shuffle, batch, prefetch)

U-Net Recap (10 min)

Architecture

- Encoder → Bottleneck → Decoder + Skips
- 3×3 conv blocks, MaxPool down; TransposedConv up
- Output: 1×1 conv with sigmoid (binary)

Loss & Metrics

- Combined: BCE + Dice (robust to imbalance)
- Metrics: IoU, Dice, Precision/Recall

Training (25–35 min)

Settings

- Optimizer: Adam (1e-4)
- Epochs: 30–50 (EarlyStopping)
- Callbacks: Checkpoint (best IoU), ReduceLROnPlateau

Monitor

- Loss, Dice, IoU (train/val)
- Stop when val metrics plateau

Evaluation (15 min)

On test set

- Report Loss, Acc, Dice, IoU
- Compute Precision/Recall/F1 (flood class)

Confusion matrix

- TN, FP, FN, TP
- Discuss precision vs recall trade-off for DRR

Visualization (10 min)

Side-by-side

- SAR VV (normalized)
- Ground truth mask
- Predicted probability map
- Error overlay (TP green, FP red, FN yellow)

Export & GIS (10 min)

Save

- Model (`.h5` / `.keras`)
- Predictions (`.npy` or GeoTIFF)

Vectorization (concept)

- `rasterio.features.shapes` → polygons
- Export GeoJSON/Shapefile for QGIS

Troubleshooting (5–10 min)

- OOM → reduce batch, mixed precision, clear session
- Flat metrics → check normalization/labels, LR, loss
- Overfitting → stronger aug, dropout, fewer filters
- All-zeros/ones → sigmoid, threshold tuning, Dice loss
- Colab disconnects → frequent checkpoints, Drive save

Time Plan

Block	Minutes
Setup + Exploration	25
Preprocessing	20
U-Net Recap	10
Training	35
Evaluation	15
Visualization	10
Export + GIS	10
Troubleshooting	10
Buffer	15

Start the Lab

Open the notebook in Colab:

`day3/notebooks/Day3_Session2_Flood_Mapping_UNet.ipynb`