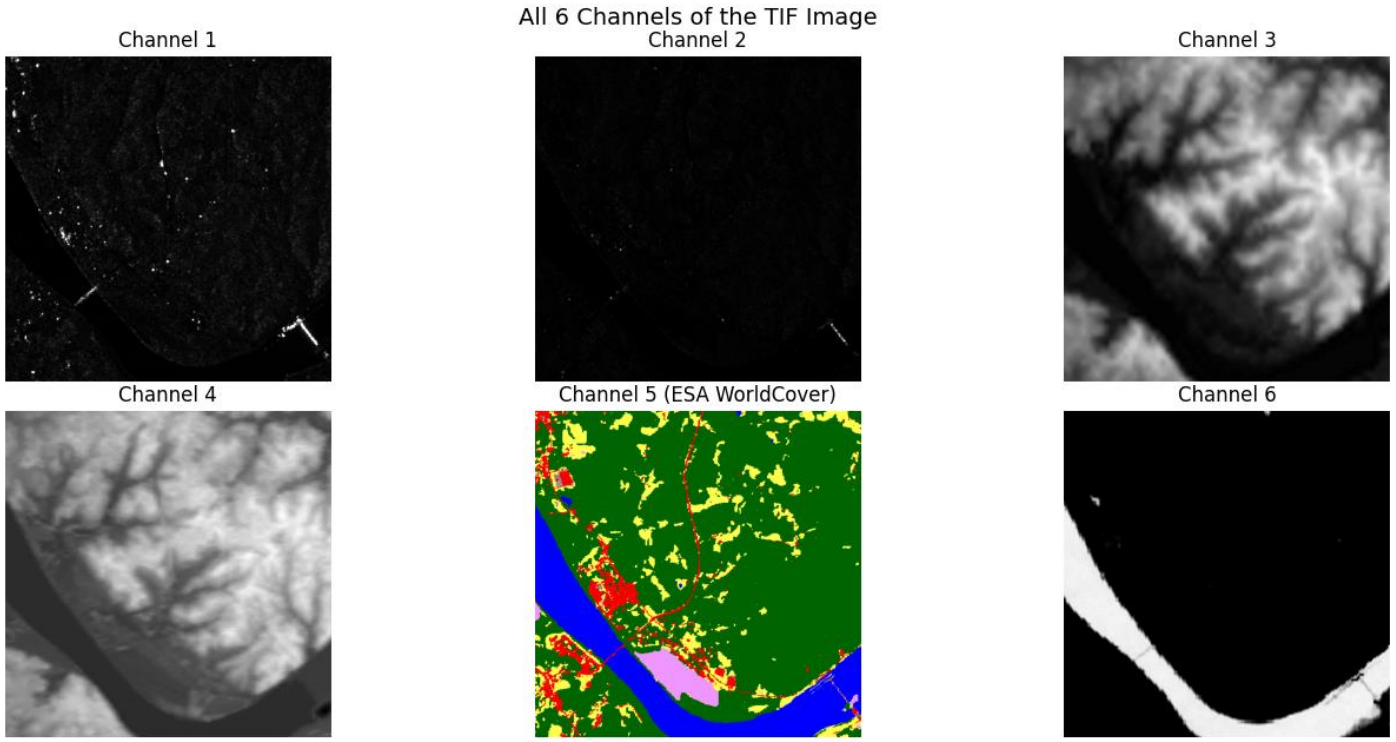
OVERCOMING THE UNCERTAINTY CHALLENGES IN FLOOD RAPID MAPPING WITH SAR DATA

He Huang[1], *Jiepan Li*[1], *Wei He*[1,†], *Hongyan Zhang*[1,2], *Liangpei Zhang*[1]

Dataset Description:

Each image has 6 channels:

- VH, VV polarizations
- Digital Elevation Models (Merit DEM & Copernicus DEM)
- Land Cover Map (ESA World Cover Map)
- Water Occurrence Probability Map



Methodology

Model used: UAFNet

Feature Extraction using Encoder-Decoder Network

- **Encoder:** Extracts **hierarchical features** from the image.
- **Decoder:** Reconstructs the **flood extraction map**.

Step 1: Encoding Features with PVT (Pyramid Vision Transformer)

- Why PVT-V2? It efficiently captures global context and spatial features.
- It processes the input image into four feature levels E1, E2, E3, E4 where E1 has high level and E4 low level features.

Step 2: Enhancing Features with Multi-Branch Dilation Convolution (MBDC)

- **Problem:** Traditional convolutions may miss fine details due to fixed receptive fields.
- **Solution:** MBDC applies dilated convolutions to capture **both fine and large-scale structures**.

Fi is the enhanced feature.

Step 3: Generating Initial Extraction Map Using Feature Pyramid Network (FPN)

Even after FPN, uncertainty exists due to:

- Noise in satellite data.
- Rare or small-scale floods that the model is unsure about

Step 4: Measuring Uncertainty Using Sigmoid Activation

- Each pixel in $M4$ has a probability (via Sigmoid function) of being **flood** or **non-flood**:
  P(x) = Sigmoid(M(x))

Uncertainty is measured as how close $P(x)$ is to 0.5:

Uf (Foreground Map) = Sigmoid(M) − 0.5 = uncertainty for flood pixels.

Ub (Background Map) = 0.5 − Sigmoid(M) = uncertainty for non-flooded pixels.

Step 5: Ranking Uncertainty Using the Uncertainty Rank Algorithm (URA)

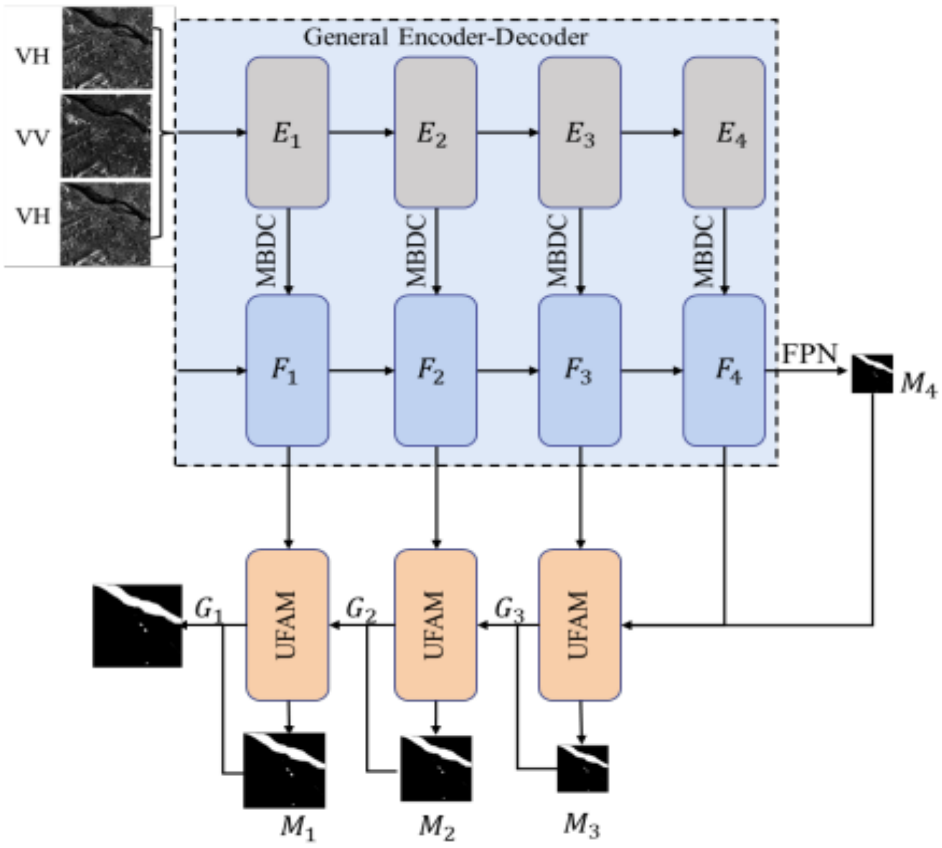
Pixels are classified into **5 uncertainty levels** based on ranges:

$$\mho(i,j) = \begin{cases} \lfloor \frac{0.5 - U_{i,j}}{0.1} \rfloor, U_{i,j} >= 0, \\ 0, U_{i,j} < 0, \end{cases}$$

Rank 5 is the Highest uncertainty whereas Rank 1 is the Lowest uncertainty.
Rank values would be used as weights.

Step 6: Fusing Features with Uncertainty-Aware Weights

- We take the highest-level feature $F4$ and the next-level feature $F3$.
- Multiply each pixel by its uncertainty rank to **highlight uncertain areas.**
- **Why?** This forces the model to **focus more on uncertain pixels** while fusing features.
- **Why upsample?** Lower-level features are smaller, so we scale them up to match higher-resolution features.
- A final convolution generates the **least uncertain** flood extraction map**.**
- Loss is calculated between the GT and the least uncertain map.


VH
VV
VH
General Encoder-Decoder
E_1
E_2
E_3
E_4
MBDC
MBDC
MBDC
MBDC
F_1
F_2
F_3
F_4
FPN
M_4
G_1
G_2
G_3
UFAM
UFAM
UFAM
M_1
M_2
M_3

IMT2021015 (Tanmay Jain) **IMT2021007(Sunny Kaushik)**

Dataset Info:

Number of images: 1631

Training data: 80% Testing Data: 20%

1. Granite Geospatial Foundation Model

- Initially Prithvi-100M designed for Image Segmentation tasks (burn scars segmentation, flood mapping, and multi-temporal crop classification).
- It was fine tuned for Flood Segmentation and Surface Water Detection tasks mainly for UK & Ireland.
- This model was further fine tuned on IGARSS 2024 dataset.

Metrics

- Overall Validation Accuracy: 0.7635
- Overall Testing Accuracy: 0.5948
- F1 Score: 0.7462
- Best Testing Accuracy after applying Filters: 0.5032

Drawbacks

- Had representations of only U.K. & Ireland flood regions.
- Didn’t incorporate the DEM resulting in a lower F1 Score.
- These problems have been solved in further work.

UNet with Attention & Uncertainty Ranking Mechanism

Model Architecture:

Encoder (Downsampling)

- Extracts deep spatial features from the input image.
- Uses convolutional layers followed by ReLU activation and Batch Normalization.
- Progressively reduces the spatial resolution while increasing the feature depth.

Bottleneck (Bridge)

The deepest layer in U-Net, having the most abstract features.

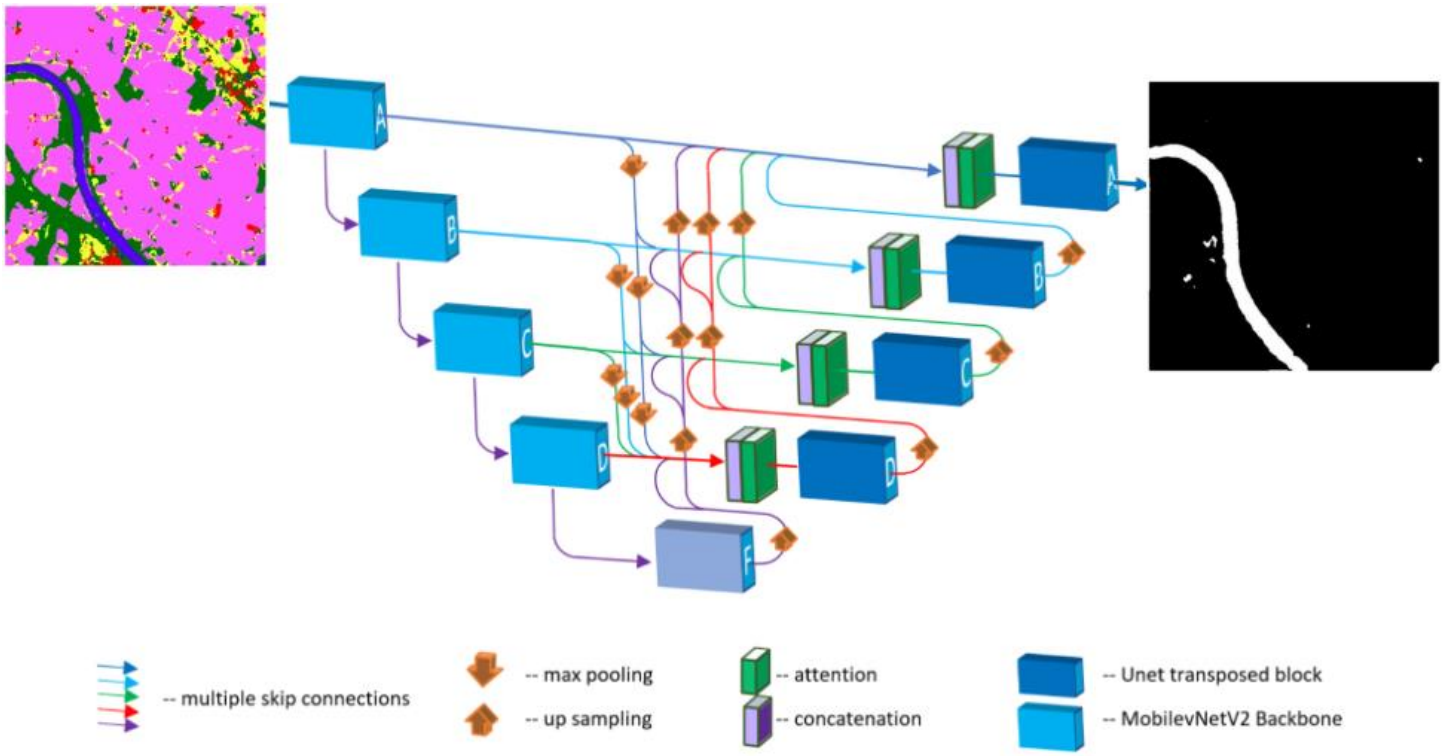
Decoder (Upsampling + Attention)

- Upsamples feature maps to restore spatial resolution.
- Uses Attention Gates to filter out irrelevant activations before concatenation.
- Helps the network focus on important regions.
- The last layer produces a segmentation probability map using a sigmoid activation.

Uncertainty Mechanism

- Computes uncertainty based on how close the probability is to 0.5.
- Weights uncertain regions higher to emphasize them in the final output.
- After calculating attention weights, the model increases the gradient updates for uncertain pixels by assigning them higher importance during backpropagation. This ensures that the network learns more effectively from ambiguous regions.

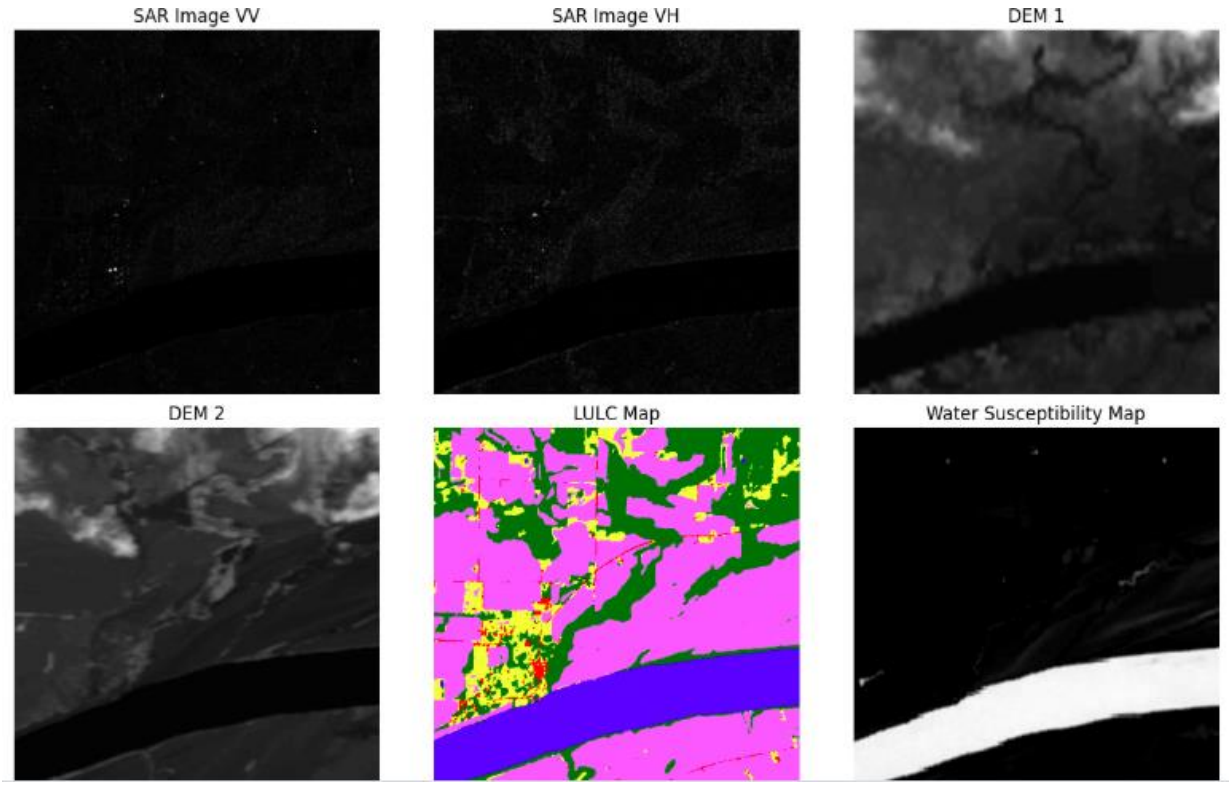
U-NET with Attention Mechanism



Inference Visualizations

All Images have 6 channels as visualized below followed by a ground truth image and a flood map predicted by the model where white pixels are flooded and black pixels are non-flooded ones.

Image 1:



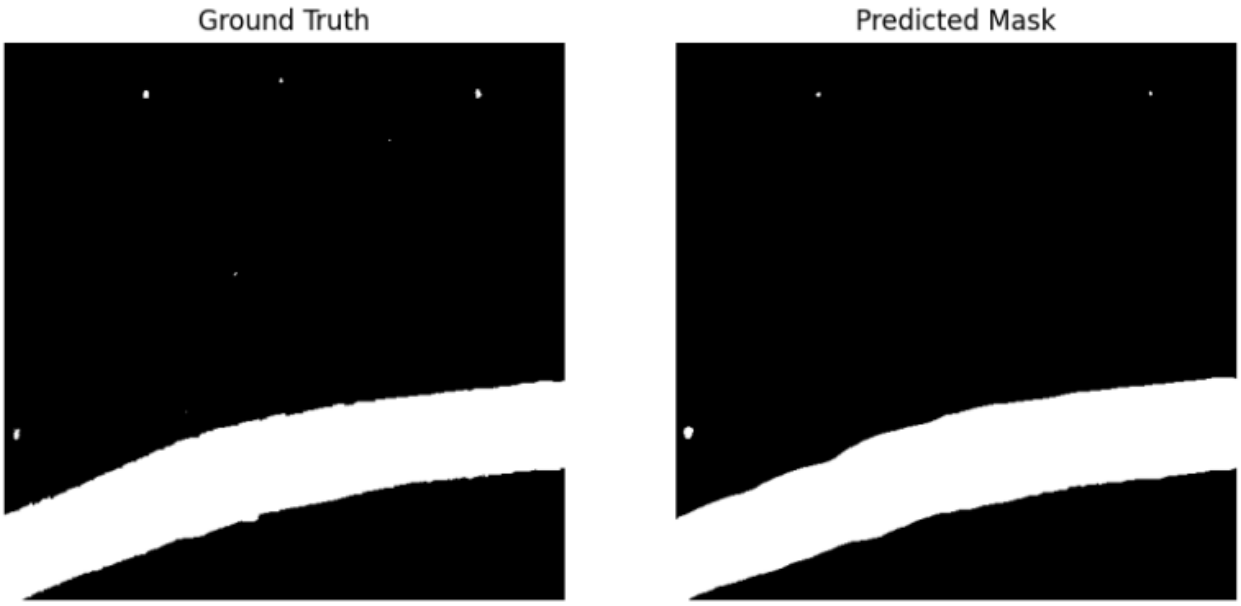
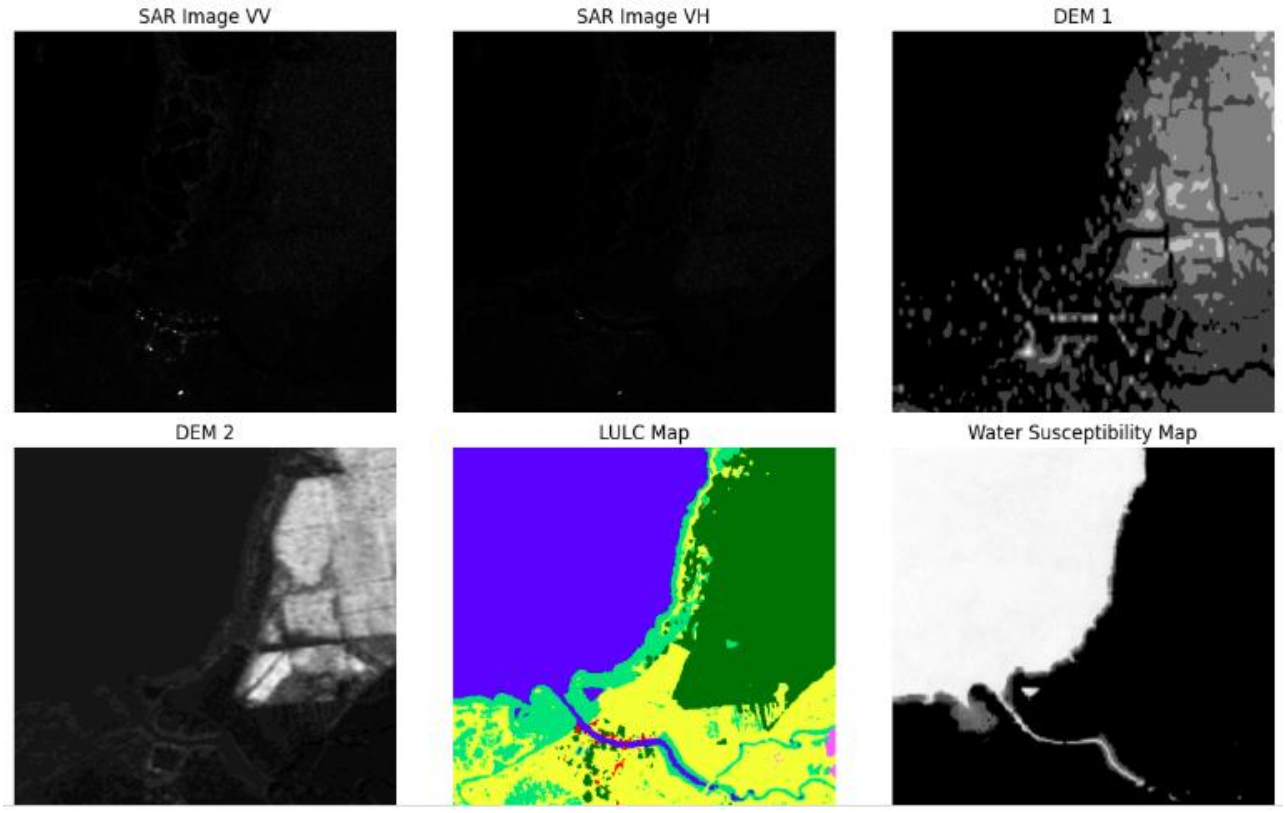

Ground Truth
Predicted Mask

Image 2:


SAR Image VV
SAR Image VH
DEM 1
DEM 2
LULC Map
Water Susceptibility Map

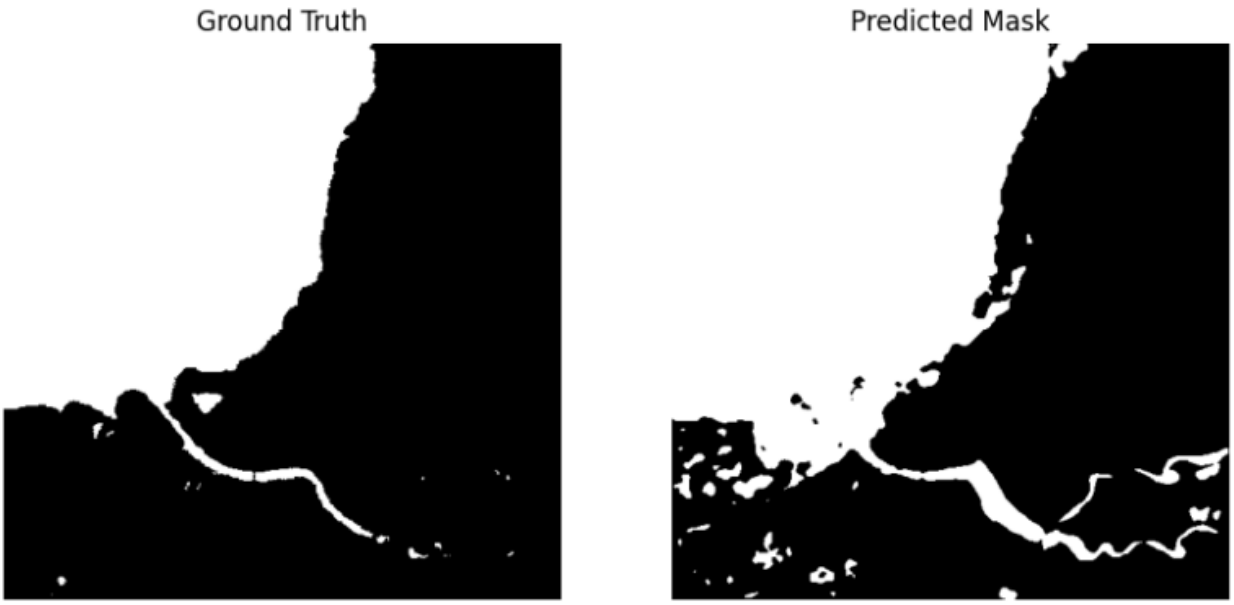
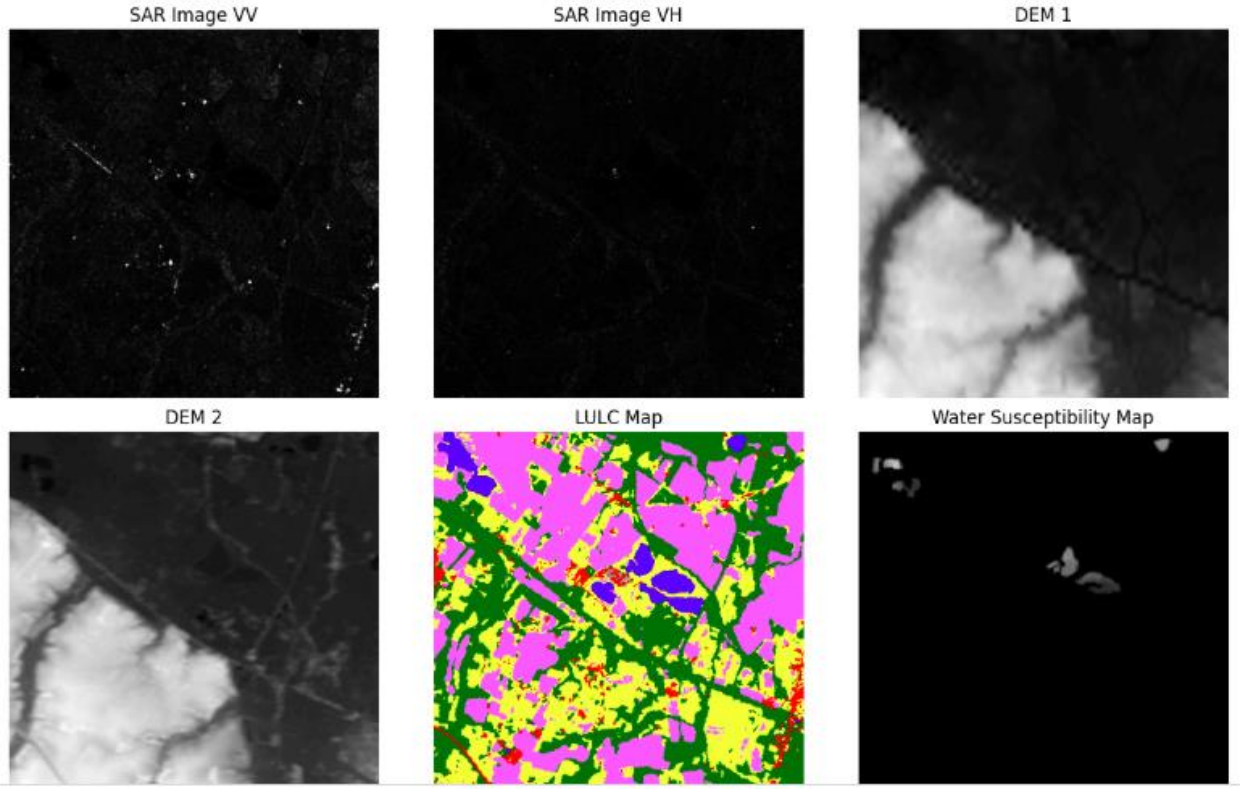
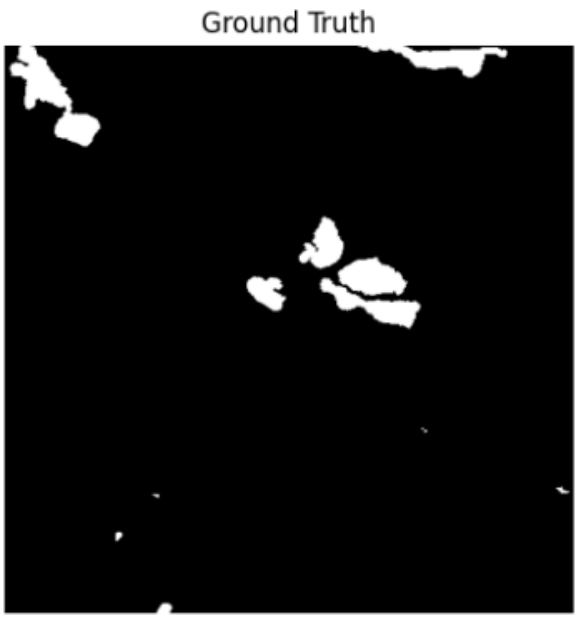
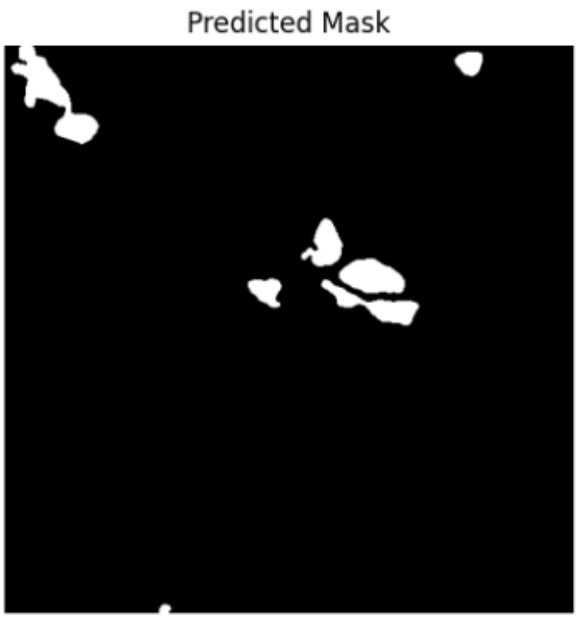

Ground Truth
Predicted Mask

Image 3:


SAR Image VV
SAR Image VH
DEM 1
DEM 2
LULC Map
Water Susceptibility Map





Metrics

- Test Accuracy: 97.52%
- Test F1 Score: 0.8413

U-NET with Attention mechanism

Tanmay Jain (IMT2021015) **Sunny Kaushik (IMT2021007)**

Fusion Technique for Multi-Channel Image Processing

1 Introduction

Fusion is performed using **FusionCNN**, a lightweight convolutional network that takes a **6-channel TIFF image** as input and outputs a **single fused feature map**. The fusion process aims to combine spatial, information to improve segmentation accuracy.

The input image consists of **6 channels**, each carrying different types of information:
Total number of channels input to **FusionCNN**:

$$C = 6$$

2 FusionCNN Architecture & Process

FusionCNN is a **convolutional network** that extracts relevant information from all six channels and **compresses them into a single feature map**.

Fusion Process

First, all the images are scaled to 10m pixel resolution using **bicubic interpolation**. Let the input tensor be X of shape (C, H, W), where $C = 6$. The fusion is done using a series of **convolutional layers**:

1. **Initial Convolution:** Extracts low-level spatial and spectral features.

$$F_1 = \text{ReLU}(\text{Conv}_{3\times3}(X))$$

   - **Input:** (C, H, W)
   - **Output:** $(32, H, W)$

2. **Feature Refinement:** Reduces redundant information and captures patterns.

$$F_2 = \text{ReLU}(\text{Conv}_{3\times3}(F_1))$$

   - **Input:** $(32, H, W)$
   - **Output:** $(16, H, W)$

3. **Final Fusion Layer:** Compresses into a **single-channel** fused feature map.

$$F = \text{Sigmoid}(\text{Conv}_{1\times1}(F_2))$$

   - **Input:** $(16, H, W)$
   - **Output:** $(1, H, W)$

The final output F is a **grayscale fused representation**, containing important **spatial, spectral, and auxiliary** information from all channels.

This **fused feature map** is then fed into **AttentionUNet** model for segmentation.

3 Accuracy metrics

- Training Accuracy: 82.38%
- Test Accuracy: 79.38%
- Test F1 Score: 0.742

4 Resources Followed and Further Plans

To focus on different better fusion strategies.

- Resource 1
- Resource 2

Weekly Report

Tanmay Jain (IMT02021015) **Sunny Kaushik (IMT2021007)**

1. Feature Fusion Model

The FusionCNN model is designed to extract, enhance, and integrate multiple feature maps from a multi-channel remote sensing image. This feature fusion process allows the model to combine low-level and high-level spatial features, which can significantly improve segmentation performance, particularly in flood detection scenarios. The final fused output is then passed to a segmentation network (such as Attention U-Net) for precise flood prediction. FusionCNN consists of multiple stages:

1. **Input Processing:** The model accepts an input tensor which includes features such as SAR images, DEM, Water Occurrence, and One-Hot Encoded LULC. The first entry convolution layer transforms the input into a higher-dimensional feature space.

2. **Multi-Scale Feature Extraction:** using Convolutional Blocks The model contains multiple convolutional blocks that progressively extract multi-scale spatial features from the input data.

   Each block consists of 2 **Standard Convolution Layers** (3×3 kernel, BatchNorm, ReLU) for feature extraction.

   Dilated Convolution Layer (Dilation = 2) to expand the receptive field and capture large-scale contextual information without increasing the number of parameters significantly.

   The model concatenates feature maps from each block, forming a rich hierarchical representation of the input data.

3. **Feature Attention Mechanism:** The extracted feature maps are passed through an attention module, which uses 1×1 convolutions to compute a refined representation of the features and applies a sigmoid activation function to generate attention weights.

   These attention weights are element-wise multiplied with the original feature maps, allowing the model to focus on the most important regions (e.g., flood-prone areas).

4. **Feature Fusion and Normalization:** The fused feature map is passed through a fusion convolution layer (3×3 kernel) to integrate attended features and dropout (0.3) to prevent overfitting. Layer Normalization to stabilize training and avoid internal covariate shifts.
5. **Final Output:** A final 3×3 convolution layer generates a single-channel feature map representing flood-prone regions. The output shape is (batch_size, H, W), which can be directly used for segmentation tasks.

2. Dice Loss

Handling Class Imbalance using Dice Loss Flood segmentation is a highly imbalanced problem, where the flooded areas (positive class) are significantly smaller than the non-flooded areas (negative class). Standard losses like Binary Cross-Entropy (BCE) tend to be biased toward the majority class, leading to poor segmentation performance in the minority class.

How Does Dice Loss Work?

Dice Loss is based on the idea of overlap measurement. It considers both the correctly predicted flood pixels (true positives) and the incorrectly predicted pixels (false positives and false negatives). Instead of simply counting the number of correct or incorrect pixels, Dice Loss looks at how well the predicted segmentation aligns with the actual segmentation as a whole region.

It pushes the model to predict segmentations that have a high overlap with the ground truth mask. It penalizes cases where the model misses flooded regions (false negatives) or wrongly classifies non-flooded areas as flooded (false positives).

Treats Small and Large Regions Fairly: Unlike BCE, which may ignore small flooded areas due to their lower pixel count, Dice Loss ensures that even small flood regions contribute significantly to the loss calculation.

4. Metrics

Training Accuracy: 96.71
Testing Accuracy: 96.34 %
Test F-1 Score: 0.7311

Weekly Report (Sunny Kaushik-IMT2021007, Tanmay Jain-IMT2021015)

The primary goal of this project is to **predict flooded pixels** in satellite imagery using a **fused input of multispectral bands and land use/land cover (LULC)** information. We make use of the **ESA WorldCover LULC dataset** and multispectral bands extracted from a time-series of TIFF images (~1600 in total). Each image contains 6 channels, where the **5th channel represents the LULC map**.

```
# ESA classes & default colors

esa_worldcover_colors = {

    10: (0, 100, 0),        # Tree cover

    20: (255, 187, 34),     # Shrubland

    30: (255, 255, 76),     # Grassland

    40: (240, 150, 255),    # Cropland

    50: (250, 0, 0),        # Built-up

    60: (150, 150, 150),    # Bare / Sparse vegetation

    70: (255, 255, 255),    # Snow and Ice

    80: (0, 0, 255),        # Water

    90: (0, 207, 117),      # Wetland

    95: (0, 168, 89),       # Mangroves

    100: (255, 255, 255),   # Moss & Lichen

}
```

📊 ESA WorldCover LULC Class Frequencies Across Dataset:

Class 10 (Tree cover): 126,365,016 pixels (29.56%)
Class 20 (Shrubland): 738,010 pixels (0.17%)
Class 30 (Grassland): 93,597,365 pixels (21.89%)
Class 40 (Cropland): 145,640,012 pixels (34.06%)
Class 50 (Built-up): 34,857,036 pixels (8.15%)
Class 60 (Bare / Sparse vegetation): 2,395,884 pixels (0.56%)
Class 70 (Snow and Ice): 135,736 pixels (0.03%)
Class 80 (Permanent Water Bodies): 18,345,145 pixels (4.29%)

Class 90 (Herbaceous Wetland): 1,853,917 pixels (0.43%)
Class 95 (Mangroves): 0 pixels (0.00%)
Class 100 (Moss & Lichen): 220,677 pixels (0.05%)

Originally, one-hot encoding each of the 11 LULC classes yielded 11 binary feature maps, creating a **16-channel input tensor** when concatenated with the 5 multispectral bands.

To improve memory efficiency and reduce model complexity:

- The 11 binary LULC maps were **collapsed into a single LULC label map** (with values from 10–100).

- The final input to the flood prediction model was reduced to **6 channels**, as all the 11 LULC channels were merged to create a new continuous image.

To visualize LULC class frequencies across the entire dataset, we created a **continuous color map** overlaid with a **heatmap gradient** representing pixel frequency. This offered insights into dominant land classes and class imbalance across the entire region.

As already discussed it won't be improving the accuracy but to just experiment we have used this approach.

Each 512×512 image is represented as a **6-channel input tensor**, ready for input to a deep learning model to perform **pixel-wise flood prediction**. This fused representation retains both **spectral information** and **terrain-type context**, increasing the robustness of flood classification under different terrains.

Metrics:

Test Accuracy: 0.8113

Test F1 score: 0.729

—----------

Plan for the next week:

Since most of the images in the LULC ESA worldcover didn’t have all the classes we have to find a way to deal with image fusion when the representation is sparse.

Following which we are looking forward to read and implement the research paper:

https://www.sciencedirect.com/science/article/abs/pii/S1746809421007370