Machine Learning (CCAI-323)
Project Report

AQUAINTEL

A COMPUTER VISION-BASED FLOOD RISK ASSESSMENT SYSTEM

Prepared by

-Joori Shareef 2212523
-Sarah Aldarwish 2210784
-Jana Alzobidi 2211632
-Jood AlHarbi 2211782
-Rahaf AlHothali 2211019

جامعة جدة
University of Jeddah

1. INTRODUCTION

Floods are One of the destructive natural disasters in the world. In the past 2 decades , floods have impacted over 2 billion people worldwide, causing extensive harm to public safety, livelihoods, and infrastructure, according to the World Health Organization. Climate change and rapid urbanization have led to a significant increase in the frequency and intensity of flooding in recent years, particularly in areas where risk mitigation planning has lagged infrastructure development.The city of Jeddah, for example, has experienced several major flood events over the past two decades causing loss of life, traffic paralysis, and significant infrastructure damage. This highlights the urgent need for localized, proactive flood response systems that can operate effectively in real world conditions.

In the immediate aftermath of a flood, one of the most critical challenges for emergency responders is determining which areas are most at risk and require urgent attention. This is not simply a matter of locating flooded zones, but of prioritizing interventions in places where human life, key infrastructure, or essential services are in greatest danger. Delays or misjudgments in this process can lead to serious consequences, including loss of life and ineffective use of rescue resources.

Despite the availability of satellite and aerial imagery, current methods for assessing flood damage often rely on manual interpretation only, which is time consuming, inconsistent, and difficult to scale during emergencies. These limitations highlight the urgent need for smarter, faster tools that can aid in analyzing flood impact on scale and under time pressure.

This project seeks to address this need by developing an process capable of segmenting and detecting flooded areas in aerial imagery and identifying the degree of risk in each area using clustring. The goal is to aid in disaster response by saving time and labor in processing massive amounts of visual data without sacrificing accuracy and trustworthiness.

2. LITERATURE REVIEW

2.1 Paper : FloodNet (2020):

As we began exploring the field of flood scene understanding using image segmentation, we looked for existing research that could provide a strong foundation for our project. One of the most relevant and widely cited works we came across was the FloodNet (2020) paper. This research introduced a high-resolution aerial imagery dataset specifically designed to support post-flood damage analysis [1].

FloodNet contains RGB images collected using UAVs during Hurricane Harvey in 2017. Each image is pixel-wise annotated across 11 semantic classes, including flooded buildings, flooded roads, trees, vehicles, and pools. The dataset was built to help machine learning models perform segmentation and classification tasks that could aid disaster response [1].

In their study, the authors used standard supervised segmentation models like U-Net and DeepLabV3+ to test how accurately these models could detect flood-related features. Their aim was to provide benchmark performance for the FloodNet dataset. While their results were strong in terms of technical segmentation, we noticed that the paper treated flood-related classes just like any other category in the scene. There was no special attention given to the importance of identifying and interpreting flood severity, which is a key requirement in real-world emergency contexts [1].

This observati on led us to build on their work but with a more focused goal. We decided to use the same FloodNet dataset, but instead of working with all 11 classes equally, we concentrated only on Class 1 (Flooded Building) and Class 3 (Flooded Road) the two classes most directly related to actual flood impact [1].

We also used DeepLabV3 as our supervised model, just like the FloodNet paper. However, our approach was different in how we analyzed and interpreted the model's output. While their analysis focused on overall segmentation accuracy, ours looked specifically at how well the model captured flooded regions, and what we could do with that information afterward [1].

PROJECT NAME	AIM OF PROJECT	DATASET	MODEL (S) USED	PERFORMANCE	YEAR
FloodNet	Understanding post-flood scenes using aerial imagery segmentation for disaster response	FloodNet	ENet, DeepLabV3+, PSPNet	Mean intersection over union (mIoU): ENet 39.84%, DeepLabV3+ 58.61%, PSPNet 80.35%	2020

Table I

presents the FloodNet project, highlighting its objective, dataset, models used, and benchmark segmentation performance.

2.1.1 Gap We Identified

One major gap in the FloodNet paper is that although it includes flood-related classes, it doesn't treat flood segmentation as a primary objective. The segmentation models are trained and evaluated across all 11 classes equally, without paying special attention to classes most critical in real-world emergencies. In practice, emergency responders don't need to detect trees or vehicles they need to know: "Where is the flood, and how severe is it?"

In our project, although we trained on all classes, we specifically evaluated and utilized the segmentation results of Class 1 (Flooded Building) and Class 3 (Flooded Road). By narrowing our focus during the analysis phase, we made the supervised output more targeted and applicable to practical flood detection use cases [1].

2.1.2 What We Did Differently

While the original FloodNet paper treated all semantic classes with equal weight, our project took a more targeted and practical approach. After training the DeepLabV3 model on the full set of classes, we focused our evaluation and usage only on the flood-relevant outputs specifically Class 1 (Flooded Building) and Class 3 (Flooded Road).

Rather than analyzing the full scene, we filtered the model outputs to isolate only the images that contained visible flood damage. This shift in focus allowed us to reduce noise, prioritize relevant cases, and make our segmentation output far more meaningful for emergency scenarios.

By doing this, our project bridges the gap between academic segmentation results and real-world needs. It transforms the supervised model from a general scene interpreter into a focused flood detection tool, better suited for field deployment and decision-making during disasters.

2.1.3 STATE OF THE ART (SOA) IN ML MODEL IMPLEMENTATION

FloodNet Research: A High-Resolution Aerial Imagery Dataset for Post Flood Scene Understanding.

FloodNet Research Segmentation comparison for top 3 Models Enet,DeepLabv3+, PSPNet:

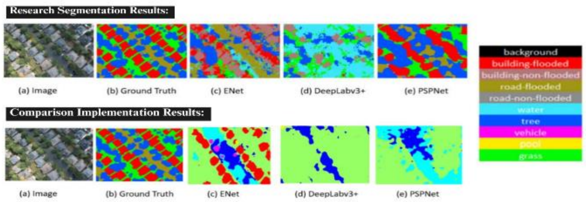

Figure 1

Visual comparison between original FloodNet segmentation results and our reproduced model outputs using ENet, DeepLabV3+, and PSPNet.

- FloodNet Research Model Performance Metric Evaluation Comparison:

Model	Results mIoU	Research mIoU
ENet	0.5140	0.426
PSPNet	0.3856	0.567
DeepLabv3+	0.4730	0.537

Table II

Comparison of mIoU scores between our reproduced results and the original FloodNet research across three segmentation models.

2.1.4 Benchmark Reproduction vs. Original FloodNet Results

The comparison between our implementation results and the original FloodNet research reveals some differences in segmentation performance across the three models. Notably, we carefully replicated the hyperparameters and training techniques described in the original paper, including the use of specific encoders, loss functions, optimizers (e.g., SGD with momentum and weight decay), learning rate schedules (PolyLR for PSPNet), and evaluation metrics (mean Intersection over Union) [1].

Despite this, our results differed slightly from those reported in the paper. ENet achieved the highest mIoU in our experiments (0.5140), outperforming both DeepLabv3+ (0.4730) and PSPNet (0.3856). In contrast, the original paper reported PSPNet as the top performer with an mIoU of 0.567, followed by DeepLabv3+ (0.537) and ENet (0.426).

One key reason for this discrepancy is the limitation in computational resources due to GPU and time constraints, we trained each model for only 10 epochs, whereas the original study likely used more extensive training. This limited training may have prevented the models, particularly PSPNet and DeepLabv3+, from reaching their full potential [1].

2.1.5 DeepLabV3 Performance Comparison: Original vs. Our Setup on Project Dataset

Miou comparison of our reproduced models (10 epochs) vs. Our final model:

ENet	PSPNet	DeepLabV3+	Our Final Model
0.5140	0.3856	0.4730	**0.625 (ours)**

Table III

compares the mIoU scores from our 10-epoch training runs of ENet, PSPNet, and DeepLabV3+ on the full FloodNet dataset, against our final DeepLabV3+ model

The table presents a comparison of mIoU scores between our reproduced models (ENet, PSPNet, and DeepLabV3+) and our final DeepLabV3+ setup. All models, including ours, were trained on the full FloodNet dataset using only 10 epochs due to computational constraints. Notably, our DeepLabV3+ model achieved the highest mIoU (0.625) on the validation set, outperforming the others. It’s important to clarify that our supervised model was trained on all 11 classes, just like the originals.

3. DATASET AND PREPROCESSING

3.1 Dataset Overview

This project uses the FloodNet dataset, a high-resolution UAV imagery collection designed for semantic segmentation of flood-affected areas. The data was captured following Hurricane Harvey (2017) using DJI Mavic Pro drones flying at 200 feet altitude, yielding a spatial resolution of approximately 1.5 cm per pixel [1].

FloodNet contains 2,343 paired samples, each consisting of an RGB aerial image and a corresponding segmentation mask. The images cover regions in Texas and Louisiana, reflecting real-world post-disaster conditions. Each mask is annotated with 10 semantic classes (e.g., flooded buildings, roads, water, vegetation), making the dataset highly suitable for flood scene understanding and analysis of urban and natural features.

The dataset is divided into three subsets:

- **Training set:** 1,445 samples
- **Validation set:** 450 samples
- **Testing set:** 448 samples

Each subset contains two folders: one for RGB images (*-org-img) and one for .png masks (*-label-img), maintaining one-to-one correspondence between inputs and labels.

3.2 Semantic Classes

Table IV lists the class labels used in the FloodNet dataset along with their corresponding integer encoding and RGB values used for visualization.

Class Label	Pixel Value	RGB Color Code
Background	0	(0, 0, 0)
Building – Flooded	1	(255, 0, 0)
Building – Non-Flooded	2	(180, 120, 120)
Road – Flooded	3	(160, 150, 20)
Road – Non-Flooded	4	(140, 140, 140)
Water	5	(61, 230, 250)
Tree	6	(0, 82, 255)
Vehicle	7	(255, 0, 245)
Pool	8	(255, 235, 0)
Grass	9	(4, 250, 7)

Table IV
Semantic Class Labels and RGB Color Codes Used in the FloodNet Dataset

3.3 Preprocessing

To standardize the input data and optimize model training, the following preprocessing and loading steps were applied to the FloodNet dataset:

- **Resizing:** All RGB images and corresponding .png masks were resized from 4000 × 3000 pixels to 256 × 256 pixels, reducing memory usage while preserving essential spatial features.
- **Normalization:** Images were normalized using standard mean and standard deviation values based on ImageNet statistics, ensuring consistent input distribution across batches.
- **Tensor Conversion:** Both images and masks were converted to PyTorch tensors using ToTensorV2(), enabling seamless integration with the deep learning framework. Masks retained their original multi-class labels (0–9) to support semantic segmentation without binarization.

All preprocessing was implemented using the Albumentations library.

To efficiently handle training and validation batches, PyTorch's DataLoader was used with the following configurations:

- **Training batch size:** 4
- **Validation batch size:** 2
- **Shuffling:** Enabled for the training set to introduce randomness
- **Workers:** 2 parallel workers to speed up loading
- **Drop Last:** Enabled to maintain consistent batch sizes by discarding incomplete batches

These steps ensured a well-structured and memory-efficient pipeline from raw data to model input.

3.4 Challenges

Some challenges encountered during preprocessing included:

- The large image resolution required significant downsampling to fit GPU memory constraints.
- The wide class imbalance (e.g., small number of pixels labeled as “vehicle” or “pool”) may impact model performance for rare classes.

3.5 Sample Visualization

To illustrate the dataset, Figure 2 shows an example of an RGB input image and its corresponding ground truth segmentation mask. The color-coded mask aligns with the semantic class labels defined in Table IV.

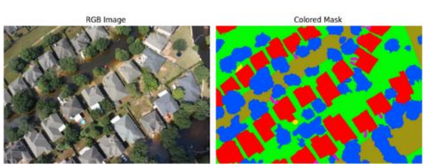

Figure 2
Example of an RGB image (left) and its annotated segmentation mask (right) from the FloodNet dataset.

4. METHODOLOGY

4.1 Supervised (DeepLabV3+)

At the start of our flood segmentation project, we set out to solve a very challenging but vital problem: how to accurately classify every pixel in satellite imagery into categories like flooded building, flooded road, vehicle, tree, and more. This required more than just choosing a model it was about finding a pipeline that could truly understand the scene pixel by pixel.

4.1.1 Trying Different Models: From ENet to U-Net to DeepLabV3+

We began our journey with ENet. We implemented it, set up our training and validation pipelines using the FloodNet-Supervised v1.0 dataset, and ran several training sessions.

To our disappointment, the results were underwhelming. The IoU hovered below 40% even after several epochs. We saw poor boundary detection and many misclassified pixels especially in complex areas like overlapping vehicles and flooded roads. We tried tuning class weights in the CrossEntropyLoss function (e.g., assigning higher weights to rare classes like flooded roads), adjusting the number of epochs, increasing the image resolution, and even testing different backbones like EfficientNet. But despite all of this, our results remained unstable.

At one point, we even trained a U-Net with a ResNet34 encoder, hoping that the skip connections and encoder-decoder architecture would improve detail recovery. While it performed better than ENet, the accuracy still plateaued around a 0.50 IoU after 15 epochs, and validation loss increased on some epochs suggesting overfitting or lack of generalization.

Eventually we moved to DeepLabV3+, and everything changed.

DeepLabV3+ is a modern encoder-decoder model built for semantic segmentation. It leverages two key innovations: Atrous Spatial Pyramid Pooling (ASPP) and dilated convolutions, which allow the model to capture both global context and local details without reducing spatial resolution . [2] This is particularly important for flood data, where small visual cues can indicate large structural differences.

We chose ResNet50 as the encoder because it balances accuracy and computational cost. It helped extract strong multi-level features from the satellite images . The decoder then refined the upsampled output to ensure sharp object boundaries. We used pretrained weights from ImageNet to accelerate convergence and improve generalization.

4.1.2 How DeepLabV3+ Processes an Image

1. **Input Preprocessing:** The image is resized, normalized, and passed to the network as a tensor .
2. **Feature Extraction (Encoder):** ResNet50 processes the image and extracts hierarchical features from low to high level.
3. **ASPP (Atrous Spatial Pyramid Pooling):** ASPP captures multi-scale features using dilated convolutions at different rates .
4. **Decoder & Upsampling:** The decoder upsamples ASPP outputs and fuses them with shallow encoder features to recover boundaries .

5. **Final Prediction:** The final segmentation map is produced with one class label per pixel, encoded as logits. [2]

4.1.3 Training Setup

We implemented DeepLabV3+ using the segmentation_models_pytorch library. Images and masks were resized to 256×256. Images were normalized and converted to tensors; masks were loaded in grayscale with class labels 0–9. We did not binarize the masks since we preserved all 10 semantic categories.

We trained the model using:

- CrossEntropyLoss: $Loss = -\sum_{i=1}^{size} y_i \cdot \log \hat{y}_i$
- Adam optimizer (learning rate: 1e-4),
- Jaccard Index (IoU) for multiclass evaluation: $\mathcal{J}(A,B) = \frac{|A \cap B|}{|A \cup B|}$

Training was conducted over 10 epochs. Batch sizes were set to 4 for training and 2 for validation. All models were trained on GPU (Kaggle backend). We saved the best model based on validation IoU.

4.2 Unsupervised

4.2.1 Previous Trial: Autoencoder-Based KMeans Clustering

To explore latent structure in the segmented masks, an unsupervised learning pipeline was first implemented using a convolutional autoencoder followed by KMeans clustering. The autoencoder was trained on binary masks highlighting only flood-relevant classes (flooded buildings and flooded roads). Once trained, the encoder component was used to extract compressed latent vectors representing spatial patterns of flooding.

For each selected latent dimension (feature_dim ∈ {8, 16, 32, 64}), KMeans was applied with k=3 to form clusters. Dimensionality reduction (via PCA) and evaluation metrics (Silhouette Score, Calinski-Harabasz Score, and Davies-Bouldin Score) were used to assess the separability and coherence of the resulting clusters.

	Silhouette Score	**Calinski-Harabasz Score**	**Davies-Bouldin Score**
Dim=8	0.5087	131.0102	0.6368
Dim=16	0.4725	67.2650	0.6921
Dim=32	0.5169	68.2344	0.6681
Dim=64	0.4756	60.3121	0.7066

Table V
Values of each evaluation metrics

While this approach successfully revealed distinguishable clusters, it had inherent limitations due to KMeans' assumption of spherical, equal-sized clusters and hard assignments. These constraints reduced flexibility and failed to capture the probabilistic or overlapping nature of real-world flood severity patterns. Moreover, when visualizing the results, the clustered regions lacked coherence—many of the grouped areas did not actually share the same level of flood risk, indicating that the assignments were not semantically meaningful.

4.2.2 Gaussian Mixture Model (GMM)

To analyze the flood severity levels in segmented images, we employed the Gaussian Mixture Model (GMM), an unsupervised learning algorithm suitable for discovering subgroups in continuous, overlapping data distributions. Unlike rigid clustering methods such as K-Means, GMM assigns probabilistic cluster memberships, which is advantageous for data where boundaries are not clearly defined.

The Gaussian Mixture Model (GMM) is a statistical approach used in unsupervised learning to discover subgroups or patterns within a dataset. It operates under the assumption that the data is composed of several underlying groups, each of which can be described by a Gaussian distribution.

GMM does not assign data points to a specific cluster in a rigid way. Instead, it estimates the probability that each point belongs to each group. This means that a single point can have partial membership in multiple groups, which offers a more flexible and realistic representation of how natural data often behaves.

To determine these probabilities and identify the groups, the model goes through an iterative process that estimates the key characteristics of each group such as its center and spread and refines these estimates by analyzing how well they explain the observed data. Over time, the model becomes better at describing the structure in the dataset.

Each group in GMM is defined by a set of statistical properties, and every data point is evaluated in relation to these groups. The end result is a probabilistic assignment of points to clusters, which reflects the model's confidence in each point's association with the different groups.

This methodology is particularly useful when the boundaries between groups are not clearly defined or when a more nuanced understanding of group membership is needed.

4.2.1 Feature Extraction

Two numerical features were extracted from each multi-class flood segmentation mask:

- **Flooded Building Ratio**: Proportion of flooded building pixels (mask == 1) to total image pixels.
- **Flood Total Ratio**: Proportion of flooded buildings and flooded roads (mask == 1 or 3) to total pixels.

These features were standardized using [insert method: e.g., z-score normalization] to ensure equal scale in clustering.

4.2.2 Clustering Process

The GMM was implemented using the [insert library, e.g., scikit-learn], configured to cluster the dataset into three components, aligning with domain knowledge suggesting three levels of flood risk: **low**, **moderate**, and **high**. The model was trained using the Expectation-Maximization (EM) algorithm.

Although the number of components was initially assumed based on practical relevance, we further validated the cluster structure using [e.g., silhouette analysis / BIC / visual inspection].

4.2.3 Interpretation

Each resulting cluster was associated with a flood severity level. The probabilistic nature of GMM allowed for soft assignment, meaning images with ambiguous severity could still be meaningfully grouped. This provided a more realistic categorization of flood risks, useful for downstream decision-making and visualization.

5. EVALUATION

5.1 Experimental Design

This study aims to assess and classify flood-affected areas using a two-stage pipeline combining supervised segmentation and unsupervised clustering. The experimental design was structured as follows:

1. **Dataset Selection and Preprocessing**
   The FloodNet-Supervised v1.0 dataset was used, consisting of 2,343 high-resolution aerial images and their corresponding annotated segmentation masks labeled with 10 semantic classes. Preprocessing steps included:

   - Resizing images and masks to 256×256 pixels
   - Normalizing RGB values using ImageNet statistics
   - Converting data to PyTorch tensors using ToTensorV2
   - Filtering outputs to include only relevant flood-related classes (Class 1: Flooded Building, Class 3: Flooded Road)

2. **Data Splitting**
   The dataset was divided into:

   - Training set (1,445 images)
   - Validation set (450 images)
   - Testing set (448 images, not used in this project phase)

3. **Feature Extraction from Predicted Masks**
   From each predicted segmentation mask, three key features were extracted:

   - Flooded Building Ratio: proportion of pixels labeled as flooded buildings.

   - Flood Total Ratio: total proportion of pixels labeled as either flooded buildings or roads.

4. **Model Selection**

   For segmentation: DeepLabV3+ with a ResNet50 encoder pretrained on ImageNet.

   For clustering: We trGaussian Mixture Model (GMM) with 3 components, chosen for its soft probabilistic clustering approach.

5. **Training Strategy**
   The model was trained for 10 epochs using:

   - CrossEntropyLoss as the loss function
   - Adam optimizer with a learning rate of 1e-4
   - Batch sizes: 4 (train), 2 (val)
   - Best model selection based on validation mIoU

6. **Evaluation Metrics**
   - Segmentation**:** Mean Intersection over Union (mIoU), with class-specific IoU for flooded building (0.6931) and flooded road (0.5353)

   - Clustering**:** Silhouette Score, with the best result being 0.6282

   - Visual Validation**:** Cluster labels were qualitatively analysed by visualizing masks per cluster.

7. **Validation & Testing**
   The trained segmentation model was evaluated on the validation set consisting of 450 images. The model achieved a mean Intersection over Union (mIoU) of 0.625, with high class-specific performance for flood-related categories:

   - Flooded Building IoU: 0.6931
   - Flooded Road IoU: 0.5353

   For the clustering stage, features extracted from predicted masks were used to classify each image into a flood risk level using a Gaussian Mixture Model (GMM). After fine-tuning the feature selection and normalization, the best Silhouette Score achieved was 0.6282, indicating good separation between the identified clusters (Low, Moderate, and Severe risk).

5.2 Results

DeepLabV3+ showed consistent improvement during training:

Epoch	Val IoU
1	0.41
5	0.57
10	0.625

Table VI
Model performance over selected training epochs

We also computed per-class IoU for the two critical flood categories:

- **class_1 (flooded building)**: 0.6931
- **class_3 (flooded road)**: 0.5353

To better understand model performance beyond metrics, we visualized a sample prediction from the validation set in Figure 3. The image below compares the input satellite image, the ground truth mask, and the predicted mask.

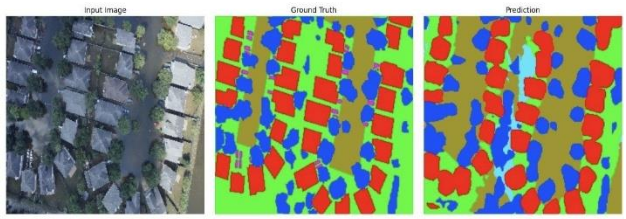

Figure3
Original flood-affected satellite image (left), its ground truth (Middle), and DeepLabV3+ prediction using ResNet50 encoder (Right)

We then extracted the features of validation images' predicted mask, Table VII shows values of extracted features for 3 random masks:

	filename	flooded_building_ratio	flooded_road_ratio
0	pred_101_0.png	0.229385	0.133469
1	pred_107_0.png	0.208740	0.164200
2	pred_84_1.png	0.001099	0.000000
3	pred_101_1.png	0.258270	0.491013
4	pred_103_1.png	0.236893	0.254471
5	pred_93_0.png	0.280502	0.254959
6	pred_100_1.png	0.225266	0.133667
7	pred_87_0.png	0.136200	0.662888
8	pred_92_0.png	0.202896	0.109756
9	pred_39_0.png	0.012100	0.000000

Table VII
The features od some masks

GGM Clustering was applied to masks using the extracted features as shown in Figure 4. This clustering achieved 0.6282 Silhouette Score.

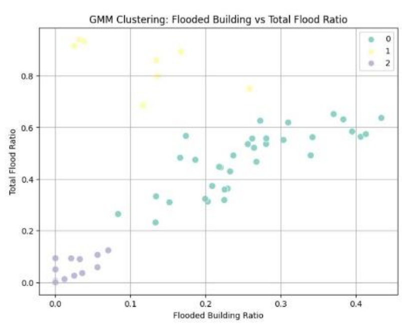

Figure 4
The result clustering on validation set

Lastly we analyzed each cluster to indicate the risk level. As concluded from result in Figure 5, Cluster 0 was associated with Low Risk, Cluster 1 was associated with Sever Risk, and Cluster 2 was associated with Moderate risk.

```
Cluster 0
flooded_building_ratio    0.039680
flood_total_ratio         0.088212
dtype: float64
------------------------------
Cluster 1
flooded_building_ratio    0.241860
flood_total_ratio         0.473856
dtype: float64
------------------------------
Cluster 2
flooded_building_ratio    0.072534
flood_total_ratio         0.832162
dtype: float64
------------------------------
```

Figure 5
analyzation of clusters

5.3 Discussion

5.3.1 Key Findings

- The DeepLabV3 segmentation model achieved good performance on the validation set with a mean Intersection over Union (mIoU) of 0.625, particularly excelling in detecting flooded buildings (IoU: 0.6931) and flooded roads (IoU: 0.5353).
- The GMM achieved a Silhouette Score of 0.6282, indicating clear separation among clusters and confirming the feasibility of risk-based grouping.
- Visual examples confirmed that cluster assignments (Low, Moderate, Severe Risk) aligned well with the actual extent of flood impact on buildings and roads.

5.3.2 Challenges

- Finding a dataset where all images came from the same satellite source, with consistent size and zoom level, was difficult. This consistency was necessary to ensure that object ratios and spatial features remained valid for segmentation and risk analysis.
- Matching clustered masks back to the original images for visual inspection and verification required maintaining filename mappings, adding extra processing steps.

5.3.3 Strengths

- The pipeline combines both supervised learning (DeepLabV3) and unsupervised learning (GMM), allowing a flexible approach to flood understanding and risk assessment.
- The selected features are interpretable and directly reflect flood exposure and infrastructure vulnerability.
- Visualization of clustered risk levels and extracted features supports post-event analysis and can aid disaster response teams in identifying high-risk regions quickly.

5.3.4 Limitations

- The approach does not currently support real-time inference, limiting its use in urgent flood response.
- The segmentation model's performance could be further improved by addressing class imbalance.

6. CONCLUSION

This project introduced an effective two stage approach to flood scene understanding by combining semantic segmentation with flood severity classification. Using DeepLabV3+, we focused on the most impactful flood related classes (flooded buildings and roads) achieving high segmentation accuracy, especially for critical areas. To assess risk levels, we extracted features from the predicted masks and applied Gaussian Mixture Modeling, successfully clustering flood severity into low, moderate, and severe categories, with a silhouette score of 0.6282.

By narrowing the model's focus and optimizing processing, our method bridges the gap between academic segmentation models and real-world emergency needs. The result is a practical and accurate system that supports rapid, informed decision-making during flood disasters.

7. References

[1] Rahnemoonfar, Maryam, et al. "FloodNet: A High Resolution Aerial Imagery Dataset for Post Flood Scene Understanding." *ArXiv.org*, 5 Dec. 2020, arxiv.org/abs/2012.02951.

[2] Chen et al., "Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation," *ECCV*, 2018. https://openaccess.thecvf.com/content_ECCV_2018/papers/Liang-Chieh_Chen_Encoder-Decoder_with_Atrous_ECCV_2018_paper.pdf

[3] MOHAMED, LABRASSI. "FloodNet." *Kaggle.com*, 2024, www.kaggle.com/datasets/labrassimohamed/floodnet. Accessed 4 May 2025.