# CMP4012

## Satellite Imaging Semester Project
## Post Flood Damage Detection


**Presented By:**

| Name | Section | Bench Number |
|---|---|---|
| Ahmed Waleed | 1 | 13 |
| Gaser Ashraf | 1 | 23 |
| Mohamed Salama | 2 | 15 |
| Yousef Mostafa ElMahdy | 2 | 42 |

**Supervised By:**

**Dr. Sandra Wahid Amin**
**Eng. Amr Abd elbakay**


**May 2023**

# Problem Definition:

Floods are one of the most devastating natural disasters that affect millions of people worldwide every year. The ability to quickly and accurately assess the damage caused by floods is crucial for emergency responders, disaster relief organizations, and government agencies to plan and allocate resources effectively. Satellite imagery provides a unique and powerful tool for post-flood damage assessment, as it can cover large areas and capture high-resolution images of the affected regions. However, manually analyzing these images is a time-consuming and labor-intensive process, making it challenging to provide timely and accurate assessments.

# Introduction:

In this project we adopted two methodologies to detect whether the image given depicts damage caused by a flood or not and also identify the area damaged.

For Classification:

1. Classical approach
    a. Random forest
    b. AdaBoost
    c. Naive Bayes
    d. XGBoost

2. Deep learning approach
    a. CNN
    b. ResNet (Pre-trained Model with fine tuning)

For images regions segmentation in the image:

1. ISODATA Clustering

# 1- Classical Approaches Pipeline:

## 1. Data preprocessing:

During this step, two operations are performed on the images. Firstly, the **images are resized** to a desired dimension. Secondly, **contrast enhancement** is applied by equalizing the histogram of each band.

## 2. Feature Extraction:

In the features extraction phase, the following techniques are employed:

1. **Modified Normalized Difference Water Index (MNDWI):** This index is calculated to identify water regions within the images.

2. **Gray-Level Co-occurrence Matrix (GLCM):** GLCM is utilized to extract texture features such as contrast, dissimilarity, homogeneity, energy (computed using graycoprops), and correlation.

3. **Color Histogram:** The color distribution of the image is captured using a histogram, providing information about the color composition.

4. **Local Binary Pattern (LBP):** LBP is employed to extract texture features based on local variations in pixel intensities.

## 3. Model Training:

Different models are used:

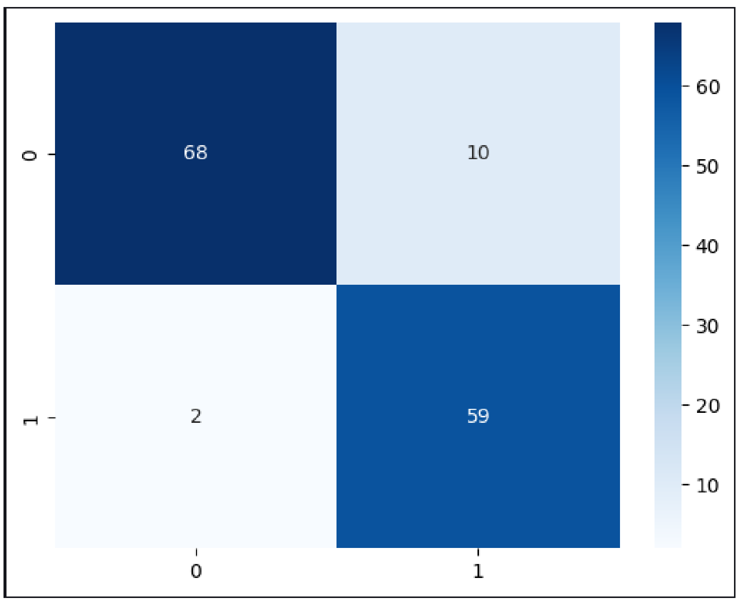
| Model | Parameters |
|---|---|
| Random Forest | n_estimators=80, max_depth=8, random_state=42 |

| | |
|---|---|
| AdaBoost | n_estimators=100 ,learning_rate=0.2 , random_state=42 |
| Naive Bayes | Default |
| XGBoost | max_depth=15 , n_estimators=150 , learning_rate=0.2, random_state=42 |

# 4. Model Evaluation:

## a. Random Forest:

1. Confusion matrix:

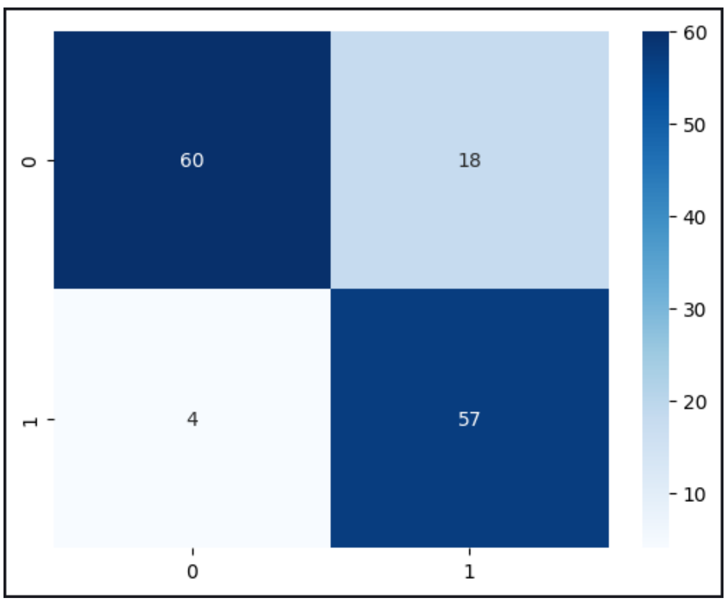



2. Precision: 0.9132505175983436
3. Recall: 0.9195039932744851
4. F1-score: 0.9133056133056132 0.9133056133056132
5. Accuracy: 0.9136690647482014 0.9136690647482014
6. Omission error: 0.08049600672551493 0.08049600672551493
7. Commission error: 0.08674948240165636

```
Evauation of Random Forest
Accuracy :  0.9136690647482014
              precision    recall  f1-score   support

           0       0.97      0.87      0.92        78
           1       0.86      0.97      0.91        61

    accuracy                           0.91       139
   macro avg       0.91      0.92      0.91       139
weighted avg       0.92      0.91      0.91       139

Precision :  0.9132505175983436
Recall :  0.9195039932744851
Ommision Error :  0.08049600672551493
Commission Error :  0.08674948240165636
F1 Score :  0.9133056133056132
Confusion Matrix :
[[68 10]
 [ 2 59]]
```

b. AdaBoost:

1. Confusion matrix:

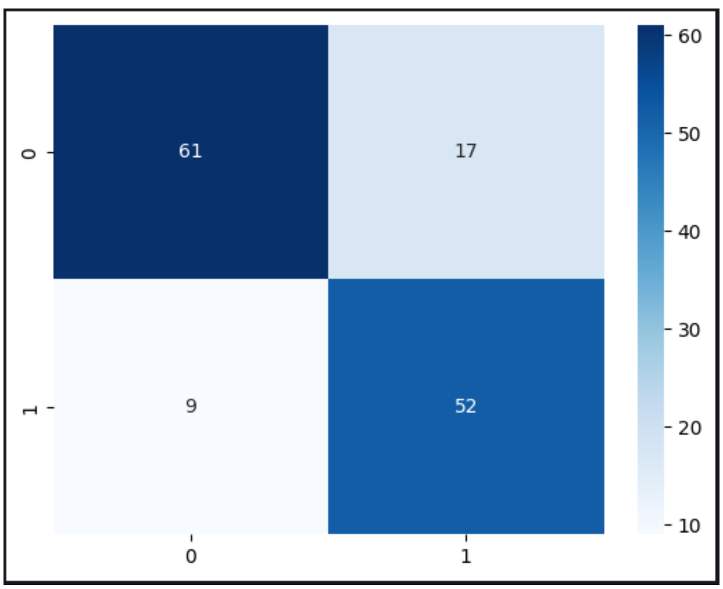



2. Precision: 0.84875
3. Recall: 0.851828499369483
4. F1-score: 0.8416528583264291
5. Accuracy: 0.841726618705036
6. Omission error: 0.14817150063051698
7. Commission error: 0.15125

```
Evauation of AdaBoost
Accuracy :  0.841726618705036
              precision    recall  f1-score   support

           0       0.94      0.77      0.85        78
           1       0.76      0.93      0.84        61

    accuracy                           0.84       139
   macro avg       0.85      0.85      0.84       139
weighted avg       0.86      0.84      0.84       139

Precision :  0.84875
Recall :  0.851828499369483
Ommision Error :  0.14817150063051698
Commission Error :  0.15125
F1 Score :  0.8416528583264291
Confusion Matrix :
[[60 18]
 [ 4 57]]
```

c. Naive Bayes:

1. Confusion matrix:

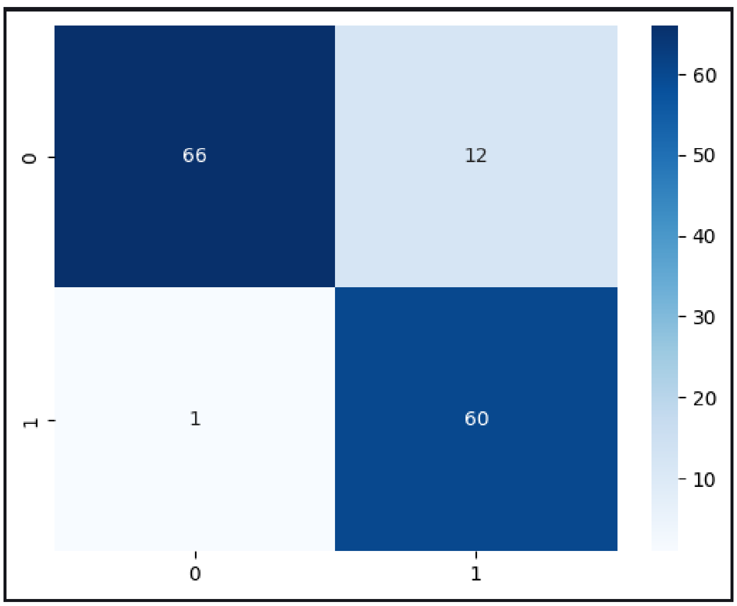



2. Precision: 0.8125258799171843
3. Recall: 0.8172551492223623
4. F1-score: 0.8121621621621622
5. Accuracy: 0.8129496402877698
6. Omission error: 0.1827448507776377
7. Commission error: 0.18747412008281572

```
Evauation of Naive Bayes
Accuracy :  0.8129496402877698
              precision    recall  f1-score   support

           0       0.87      0.78      0.82        78
           1       0.75      0.85      0.80        61

    accuracy                           0.81       139
   macro avg       0.81      0.82      0.81       139
weighted avg       0.82      0.81      0.81       139

Precision :  0.8125258799171843
Recall :  0.8172551492223623
Ommision Error :  0.1827448507776377
Commission Error :  0.18747412008281572
F1 Score :  0.8121621621621622
Confusion Matrix :
[[61 17]
 [ 9 52]]
```

d. XGBoost:

1. Confusion matrix:




2. Precision: 0.9092039800995024
3. Recall: 0.9148802017654476
4. F1-score: 0.9063002333419756
5. Accuracy: 0.9064748201438849
6. Omission error: 0.0851197982345524
7. Commission error: 0.09079601990049757

```
Evauation of XGBoost
Accuracy :  0.9064748201438849
              precision    recall  f1-score   support

           0       0.99      0.85      0.91        78
           1       0.83      0.98      0.90        61

    accuracy                           0.91       139
   macro avg       0.91      0.91      0.91       139
weighted avg       0.92      0.91      0.91       139

Precision :  0.9092039800995024
Recall :  0.9148802017654476
Ommision Error :  0.0851197982345524
Commission Error :  0.09079601990049757
F1 Score :  0.9063002333419756
Confusion Matrix :
[[66 12]
 [ 1 60]]
```

# 2- Deep Learning Approaches:

1. Data preprocessing:

During this step, two operations are performed on the images. Firstly, the **images are resized** to a desired dimension. Secondly, The images are converted from **RGB to BGR**, then each color channel is **zero-centered** with respect to the ImageNet dataset, without scaling.

Another secondary preprocessing which was applied for the CNN model training data is data augmentation which applies a diverse set of transformations on the input image which has helped the model to generalize better and increased the size of the dataset.

2. Feature Extraction:

Image themselves are the features flattened to a vector of size W*H*3.

3. Model Training:

a. CNN:

| Epoches | Batch Size |
| --- | --- |
| 50 | 64 |

loss=categorical_crossentropy | optimizer=adam

Total params: 18,146,178
Trainable params: 18,145,346
Non-trainable params: 832

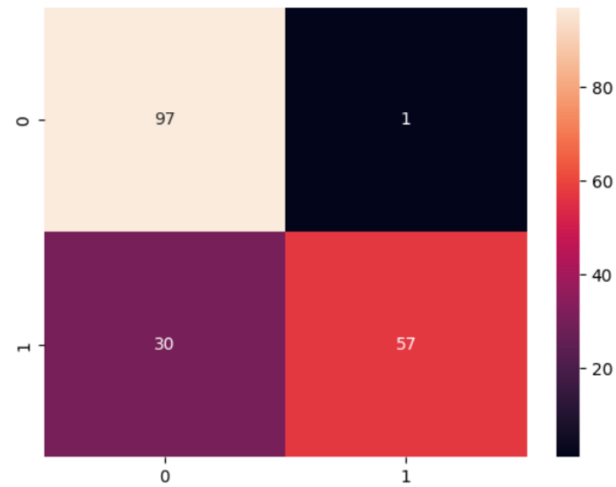
b. ResNet:

ResNet-50 is a deep convolutional neural network architecture with 50 layers. It introduces residual connections to address the vanishing gradient problem, gradually reduces spatial dimensions, and applies global average pooling for classification. It has achieved state-of-the-art performance in computer vision tasks. Trained on imageNet dataset.

## 4. Model Evaluation:

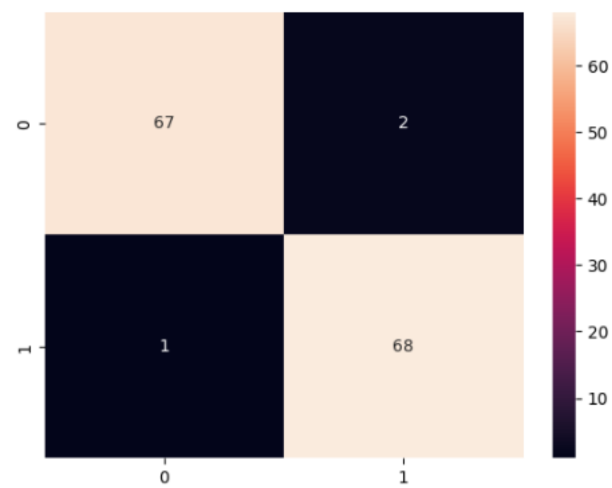
1. CNN:

    a. Precision: 0.873
    b. Recall: 0.822
    c. F1 score: 0.824
    d. Accuracy: 0.832
    e. Omission error: 0.177
    f. Commission error: 0.126




2. ResNet:

    a. Accuracy: 0.9783
    b. F1 score: 0.9783
    c. Precision: 0.9784
    d. Recall: 0.9783
    e. Omission error: 0.0217
    f. Commission error: 0.0216




## Choosed Model:

- ResNet50
- Reason: ResNet (Residual Neural Network) is an effective deep learning architecture for flood detection in images. It excels at capturing complex patterns and features in visual data. ResNet's advantages for this task include its deep architecture, which allows it

to handle the hierarchical nature of image data, and its ability to extract useful features from large-scale training datasets. The network's residual connections enable it to preserve lower-level information and capture fine-grained details important for flood detection. Additionally, ResNet's pretrained models can be fine-tuned on smaller flood-specific datasets, enabling transfer learning and improving performance with limited training samples. Overall, ResNet is a powerful tool for analyzing images, extracting relevant features, and classifying whether an image contains indications of a flood.

## 3- Images Regions Segmentation:

### 1. ISODATA Clustering:

- Max clusters: 2 (as all we need are water class and other class)
- Max iterations: 100
- Minimum distance: 20
- Minimum cluster size: 10

### 2. K Means Clustering:

- K = 2

Example output: