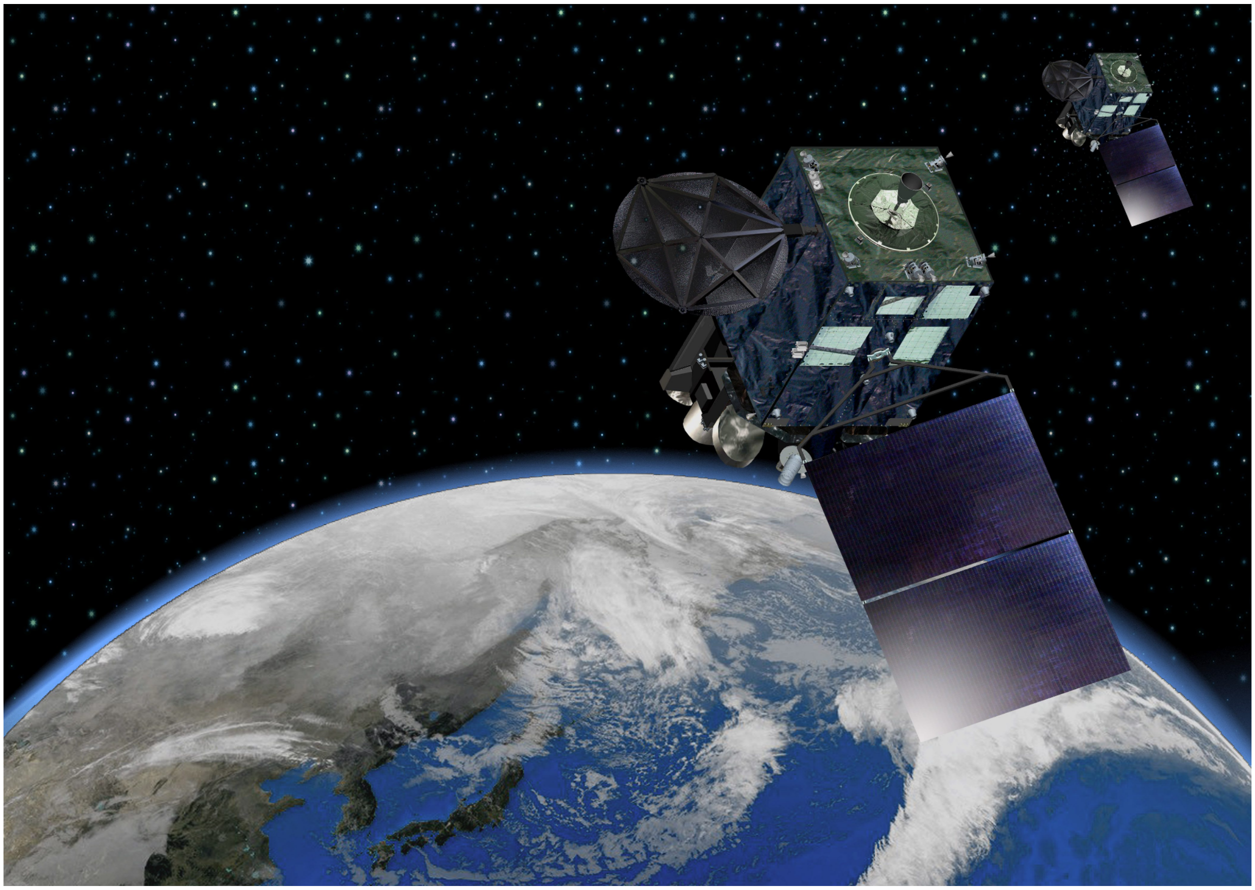


Satellite Imaging Project

Team 20

Name	Sec	B.N.
Reem Emad	1	33
Osama Magdy	1	14
Ziad Atef	1	35
Yousef Gamal	2	39

Introduction

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.

The goal of this project is to develop an automated method for detecting post-flood damages in satellite imagery using machine learning and computer vision techniques. The proposed solution will allow for the rapid and accurate identification of damaged areas, enabling emergency responders to quickly prioritize their response efforts and allocate resources more effectively. The proposed solution can also be used to monitor the long-term impact of floods on the affected areas, allowing for more informed decision-making and better disaster preparedness in the future.

In summary, the proposed solution has the potential to make a significant impact in disaster response and recovery efforts, ultimately improving the lives of people affected by floods.

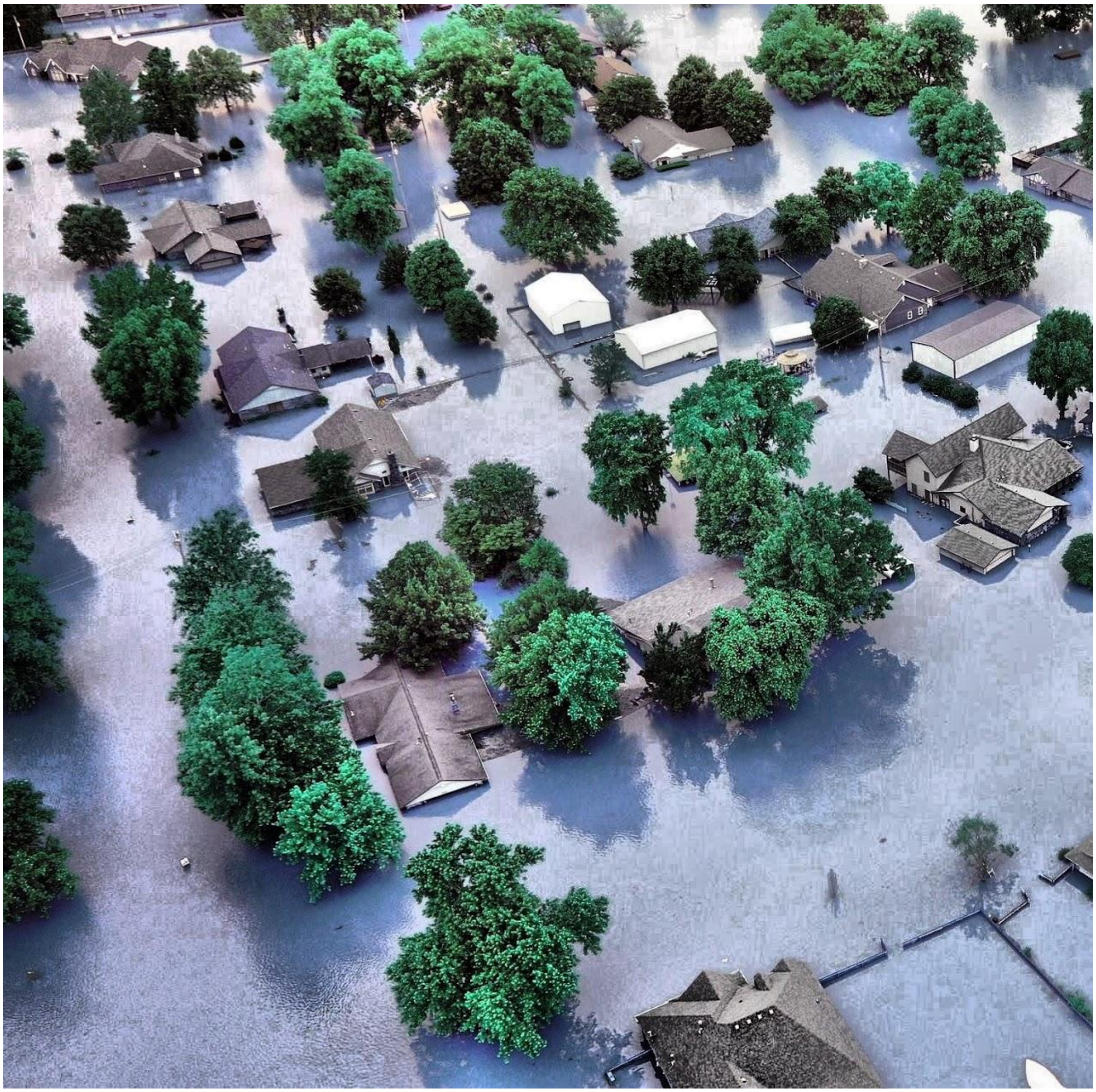
Project pipeline

The pipeline is split into three main stages:

Assume we have this image:

Data preprocessing:

1. Contrast Enhancement: As the problem is close to the segmentation problems we found it's a good choice to perform contrast enhancement over all images to distinguish better between water regions and land regions (just preprocessing)



2. Resizing Images: All images are from different sizes which is not suitable for many cases in the learning algorithms and even feature extraction. Resizing each image to 256*256

3. Data Augmentation: We noticed that the data size is small and not much suitable for learning, especially deep learning techniques. Data augmentation is done by rotating each image 90 degrees and 180 degrees. Thus, increasing the size 2 times more.

Feature Extraction:

1. Indexes: Following indexes will be estimated depending on the RGB image channels
    a. Normalized Difference Water Index: Calculated using Green & Blue channels
    b. Normalized Difference Vegetation Index: Calculated using Green & Red channels
2. Color features: Color variance significantly differs between flooded and non-flooded images. Difference in blue color value between 2 flooded images is too small, while this difference between flood and non-flooded images is large. So we tried following color features
    a. Color averaging: Getting average color of the image
    b. Color variance: Calculate variance in color across image
    c. Color histogram: generating image color histogram

    Results found was as follows:

    - The difference in mean color between flooded and non-flooded images in the blue channel is 9.356713619725781
    - The difference in color variance between flooded and non-flooded images in the blue channel is 2318.8324618520296
    - The difference in mean color between the 2 flooded images in the blue channel is 0.7572478863561116
    - The difference in color variance between the 2 flooded images in the blue channel is 2810.911136979508
3. Texture features: Texture energy and texture homogeneity have slight difference with 2 flooded images, while it differs with the flooded and non-flooded images
    a. Texture gradient: Feature that describes how quickly the texture changes in an image. It is calculated by measuring the differences in pixel intensity between neighboring pixels and can be used to characterize the roughness or smoothness of a surface in an image.
    b. Texture energy: Texture energy describes the overall amount of texture in an image. It is calculated by summing up the squared values of the local differences in pixel intensity across the entire image. A high texture energy

value indicates that the image contains a lot of textured areas, while a low value corresponds to smooth areas.

c. Texture correlation: Texture correlation measures the similarity between different parts of an image in terms of their texture. It is calculated by comparing the pixel values of two patches of the same size at different locations in the image. A high correlation value means that the two patches have similar textures, while a low correlation value indicates that they have different textures.
d. Texture homogeneity: Texture homogeneity measures the similarity of the texture within a small patch of an image. It is calculated by computing the variance of the pixel values within the patch. A low variance indicates that the patch has a more uniform texture, while a high variance indicates that the texture varies significantly within the patch.

4. Shape feature: Summation of compactness and summation of eccentricity have significant differences between flooded and non-flooded, while 2 flooded have no difference. Those features mainly depend on dividing images into regions then calculating the sum of the following over different regions.
    a. Area: getting area sum across regions
    b. Perimeter: getting perimeter sum across regions
    c. Compactness: calculating compactness using area and perimeter across regions
    d. Eccentricity: getting eccentricity sum across regions

Classification:

1. Classical classification:
    a. with texture and shape features only:
        - **adaboost -> 89%**
        - Random forest -> 86%
        - Knn -> 85%
        - logistic regression -> 75%
        - svm (rbf) -> 52%
    b. with all features summarized:
        - **Random forest -> 93%**

- Knn -> 72%
- logistic regression -> 77%
- svm (rbf) -> 68%
- adaboost -> 88%

c. PCA with all features summarized:

- **adaboost -> 72%**
- Random forest -> 70%
- Knn -> 66%
- logistic regression -> 67%
- svm (rbf) -> 70%

d. (Augmented Data) with all features summarized:

- **Random forest -> 92%**
- Knn -> 66%
- logistic regression -> 83%
- svm (rbf) -> 70%
- adaboost -> 88%

2. DL:
   a. CNN is used E2E (just preprocessing)
      - F1 score → 91%

   b. VGG16 E2E (just preprocessing)
      - **F1 score → 100%**

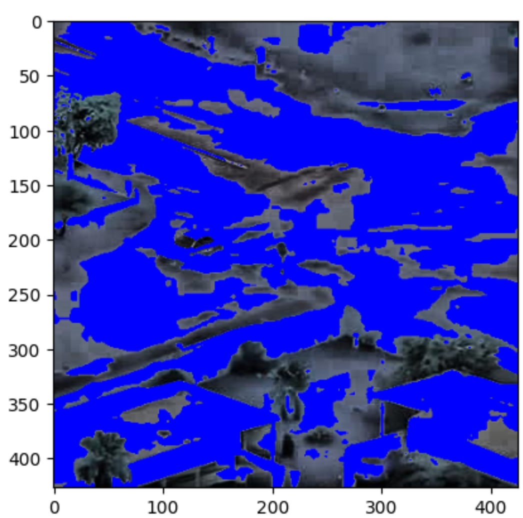
   c. We used feature extraction by deep learning techniques and classified with classical machine learning classifiers:
      - Resnet for features and logistic regression for classification: F1 score → 93.5%
      - **CNN concatenate 3 features (Resnet, InceptionV3, Xception) with logistic regression --> 98.9%**
      - CNN concatenate 3 features (Resnet, InceptionV3, Xception) with SVM --> 80%
      - CNN concatenate 3 features (Resnet, InceptionV3, Xception) with adaboost --> 91%
      - CNN concatenate 3 features (resnet, InceptionV3, Xception) with random forest 95.5%

Clustering:

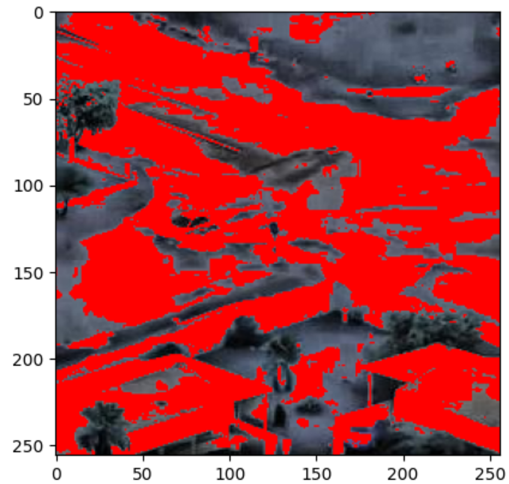
Clustering is used over all flooded images to detect which parts of the image are flooded and which are not. We used clustering techniques for segmenting the image. Assume we have this image:

Here is our results:

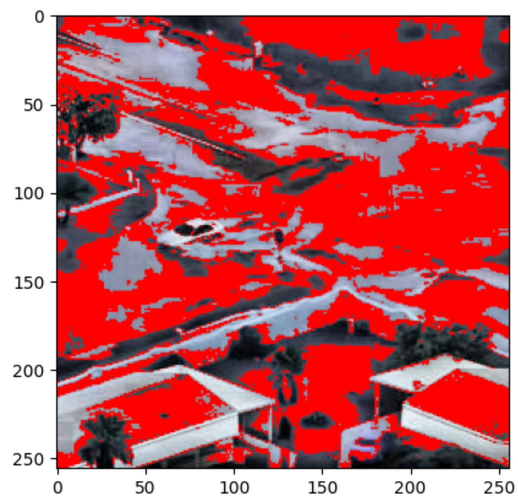
1. Binarization



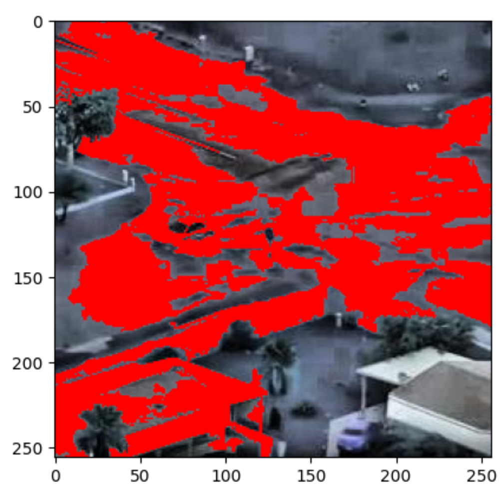
2. K means with K=2



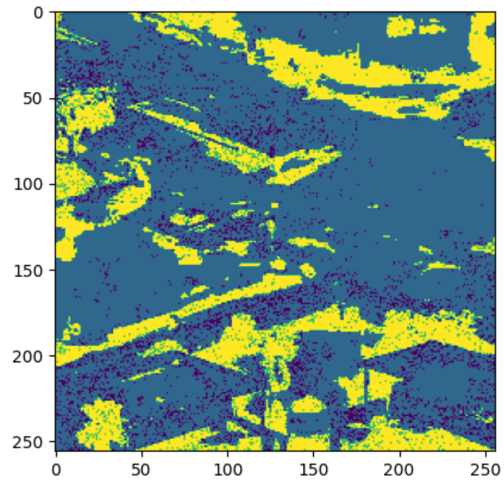
3. K means with K=3



4. Region growing



5. ISODATA



Conclusion: We selected the Region growing as it has less false positive regions (non-flooded regions classified as flooded). They all seem to have a close rate of false negatives.

Evaluation(Showing evaluation for best features)

1. Classical model:

a. Random forest:
   i. Accuracy: 0.9354838709677419
   ii. Macro F1-score: 0.9354166666666666
   iii. Omission: 0.0967741935483871
   iv. Commission: 0.03225806451612903
   v. Recall: 0.9032258064516129
   vi. Precision: 0.9655172413793104
   vii. Confusion matrix:
      1. TP: 84
      2. FP: 3
      3. TN: 90
      4. FN: 9

b. Logistic regression:
   i. Macro F1-score: 0.8387096774193549
   ii. Accuracy: 0.8387096774193549
   iii. Omission: 0.16129032258064516

iv. Commission: 0.16129032258064516
v. Recall: 0.8387096774193549
vi. Precision: 0.8387096774193549
vii. Confusion matrix:
1. TP: 78
2. FP: 15
3. TN: 78
4. FN: 15

c. SVM
   i. Macro F1-score: 0.6244758502873117
   ii. Accuracy: 0.6505376344086021
   iii. Omission: 0.08602150537634409
   iv. Commission: 0.6129032258064516
   v. Recall: 0.9139784946236559
   vi. Precision: 0.5985915492957746
   vii. Confusion matrix:
      1. TP: 85
      2. FP: 57
      3. TN: 36
      4. FN: 8
d. Adaboost:
   i. Macro F1-score: 0.8870151282867142
   ii. Accuracy: 0.8924731182795699
   iii. Omission: 0.13978494623655913
   iv. Commission: 0.08602150537634409
   v. Recall: 0.8602150537634409
   vi. Precision: 0.9090909090909091
   vii. Confusion matrix:
      1. TP: 80
      2. FP: 8
      3. TN: 85
      4. FN: 13
e. KNN::
   i. F1-score: 0.854054054054054
   ii. Accuracy: 0.8548387096774194

  iii. Omission: 0.15053763440860216
  iv. Commission: 0.13978494623655913
  v. Recall: 0.7204301075268817
  vi. Precision: 0.7282608695652174
  vii. Confusion matrix:
    1. TP: 67
    2. FP: 25
    3. TN: 68
    4. FN: 26

2. Deep Learning Models:

a. CNN concatenate 3 features (Resnet, InceptionV3, Xception) with logistic regression:
  i. Macro F1-score: 0.989247311827957
  ii. Accuracy: 0.989247311827957
  iii. Omission: 0.010752688172043012
  iv. Commission: 0.010752688172043012
  v. Recall: 0.989247311827957
  vi. Precision: 0.989247311827957
  vii. Confusion matrix:
    1. TP: 92
    2. FP: 1
    3. TN: 92
    4. FN: 1

b. VGG16:
  i. Macro F1-score: 99.4722292%
  ii. Accuracy: 99.46%
  iii. Omission: 0.0105
  iv. Commission: 0
  v. Recall: 0.9895
  vi. Precision: 1.0000
  vii. Confusion matrix:
    1. TP: 96
    2. FP: 0
    3. TN: 88
    4. FN: 1

Final Model

VGG16 (Highest Accuracy)