Details regarding the submitted Folder:

1. A ppt of the presentation, namely **GNR 602 Presentation Slides.ppt**
2. **GNR602.py**: Source file.
3. Input file namely, **"Input 1", "Input 2", "Input 3"** and **"Input 4".**
4. **GNR602.exe:** .exe file.
5. **readme.pdf:** This file.

Input x folder:

1. It contains an **image file** (the input image) and **GLMC file** and the **"Condition x"** folder where the output is stored.
2. Multiple **"Condition x "** folders are created to run the program on different parameters.

Output Folder **"Condition x"** contains:

1. **inputs.txt:** It contains information about the input & output file locations and input parameters you provided.
2. **ORIGINAL IMAGE.JPG:** The original image that was classified.
3. **GREY IMAGE.JPG:** The greyscale image.
4. **SEGMENTED IMAGE.JPG:** The segmented image (greyscale).
5. **COLOR SEGMENTED IMAGE.JPG:** The segmented image (colored).
6. **Original vs Segmented.JPG:** The original and segmented image (colored) side by side.
7. **Classified Image RF.JPG:** The RF classified image.
8. **Original vs RF Classified.JPG:** The original and RF classified image side by side.
9. **Rf_model2.joblib :** The RF model.

Steps in running the .exe file:

1. Start the GNR602.exe file and follow the instructions given in the terminal opened after starting GNR602.exe.
2. Select the Image File.
   The input image is in the folder namely, **"Input x" (x=1,2,3,4)**.
3. Select the directory where the output files are to be saved.
4. Select whether to blur the image by typing "yes" or "no".
5. Enter the number of clusters you want to segment and classify the image.
6. Enter the max number of iterations you want K-means to run.[around 10]
7. Enter the epsilon for K-means. [around 0.1 to 1]
8. Select the GLCM file based on your image.
   The GLCM files will be in the same folder as that of your input image folder namely, **"Input x" (x=1,2,3,4)**. The GLCM file name will be of the type: **"GLCM_x".**
9. Enter the number of samples per cluster.
10. Outputs will be stored in the **[Grp47_Pegu_Sikchi_Raj\Input x\Condition x].**

Note: For the input values in 4,5,6,8, you can refer to the **inputs.txt file** available in the **[Grp47_Pegu_Sikchi_Raj\Input x\Condition x]** folder or below.

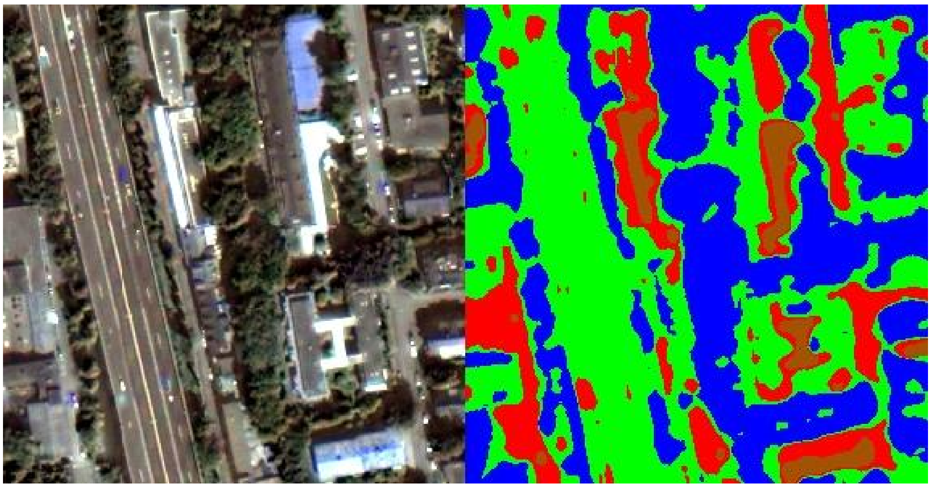
The RF accuracy and output might vary a little as the training sample is selected randomly.

Input File 1:

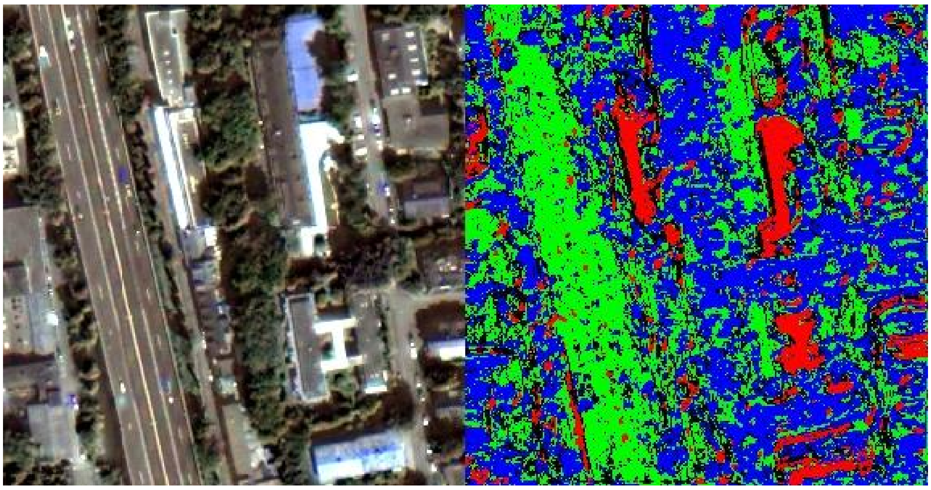
GLCM_1.1.npy: GLCM calculated for 4 directions and 4 distances and window of 5 for image 1.jpg.

Condition 1:

Image: 1.png
GLCM used: GLCM_1.1.npy
Number of pixels: 85239
Blurred: yes
Number of clusters: 4
Iterations of kmeans: 10
Epsilon of kmeans: 1.0
Number of Samples per cluster: 2000
RF accuracy: 0.581



Original vs Segmented Image



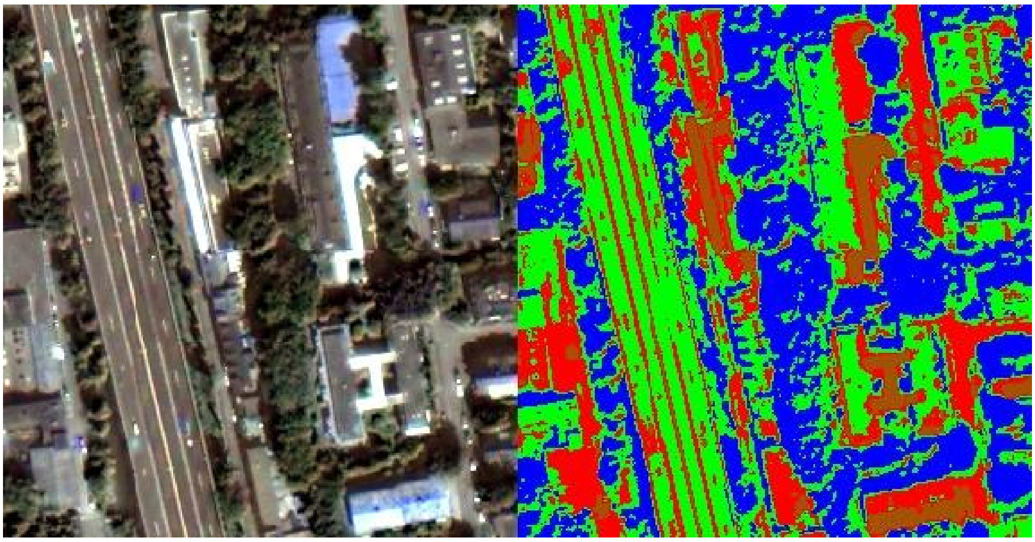
Original vs RF Classified Image

Input File 1:

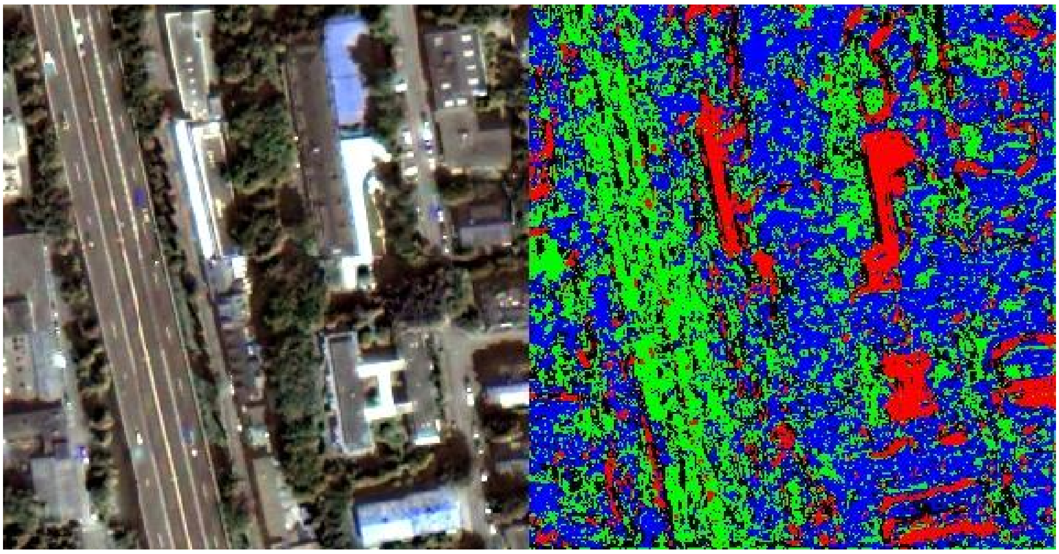
GLCM_1.1.npy: GLCM calculated for 4 directions and 4 distances and window of 5 for image 1.jpg.

Condition 2:

Image: 1.png
GLCM used: GLCM_1.1.npy
Number of pixels: 85239
Blurred: no
Number of clusters: 4
Iterations of kmeans: 10
Epsilon of kmeans: 1.0
Number of Samples per cluster: 2000
RF accuracy: 0.5145



Original vs Segmented Image



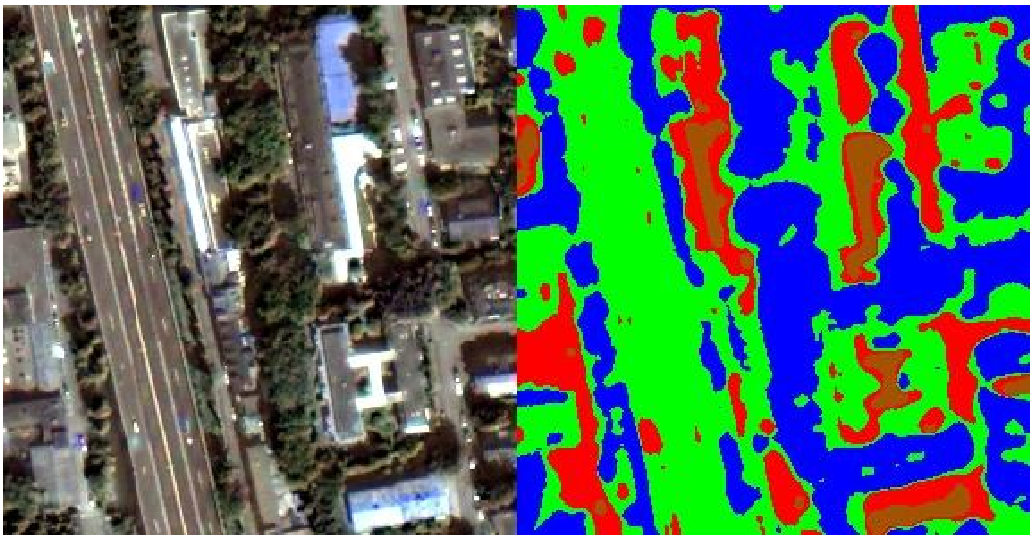
Original vs RF Classified Image

Input File 1:

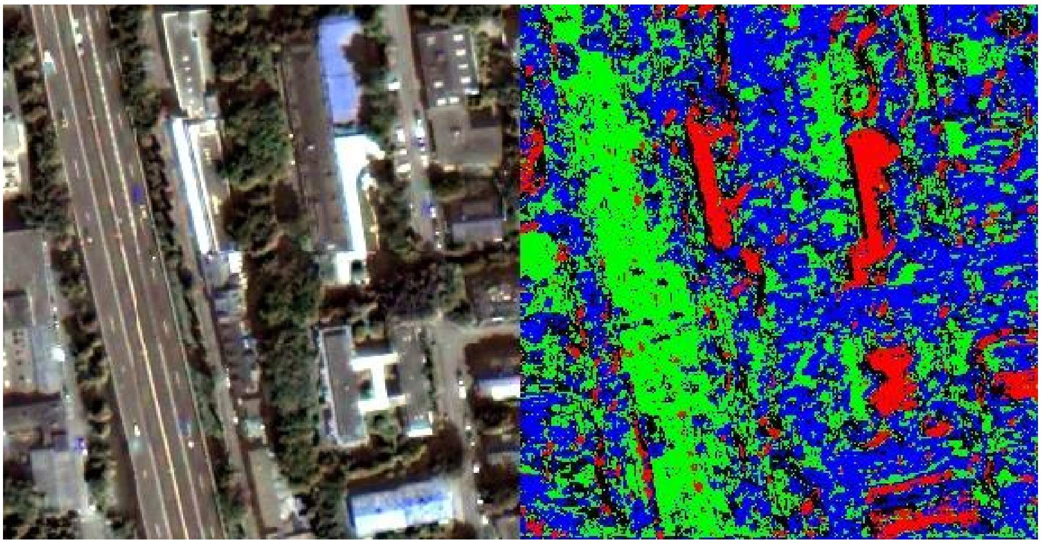
GLCM_1.2.npy: GLCM calculated for 3 directions and 3 distances and window of 5 for image 1.jpg.

Condition 3:

Image: 1.png
GLCM used: GLCM_1.2.npy
Number of pixels: 85239
Blurred: yes
Number of clusters: 4
Iterations of kmeans: 10
Epsilon of kmeans: 1.0
Number of Samples per cluster: 2000
RF accuracy: 0.5775



Original vs Segmented Image



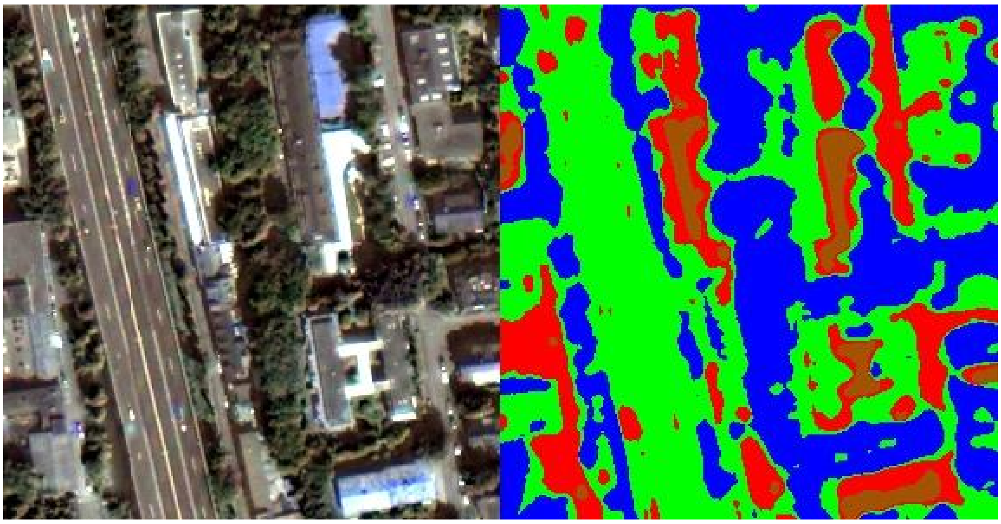
Original vs RF Classified Image

Input File 1:

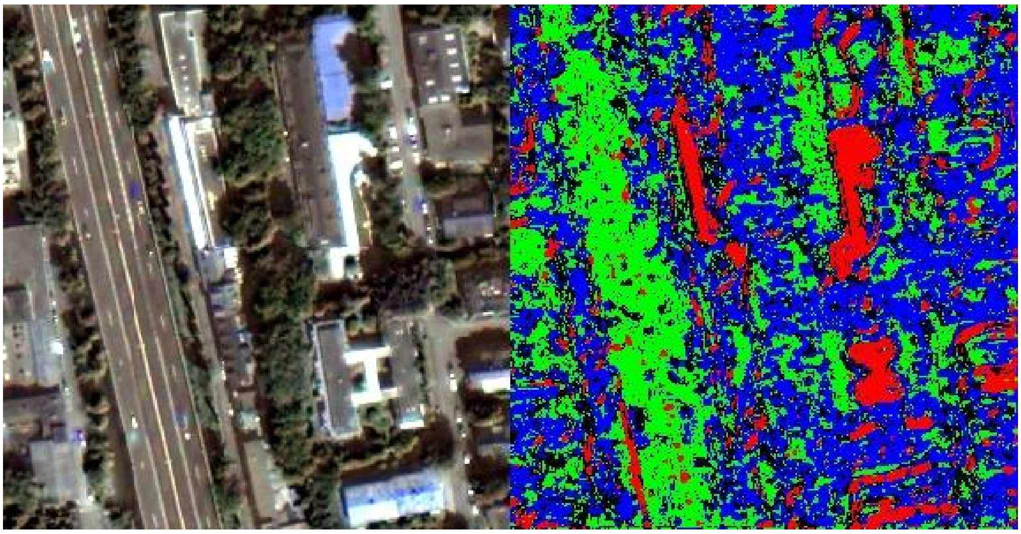
GLCM_1.2.npy: GLCM calculated for 3 directions and 3 distances and window of 5 for image 1.jpg.

Condition 4:

Image: 1.png
GLCM used: GLCM_1.2.npy
Number of pixels: 85239
Blurred: yes
Number of clusters: 4
Iterations of kmeans: 10
Epsilon of kmeans: 1.0
Number of Samples per cluster: 500
RF accuracy: 0.504



Original vs Segmented Image



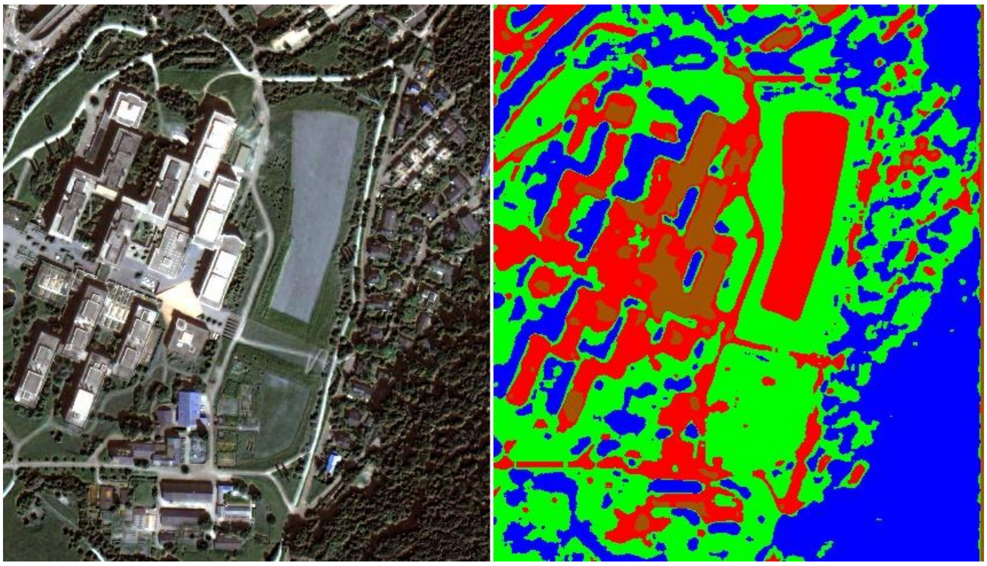
Original vs RF Classified Image

Input 2:

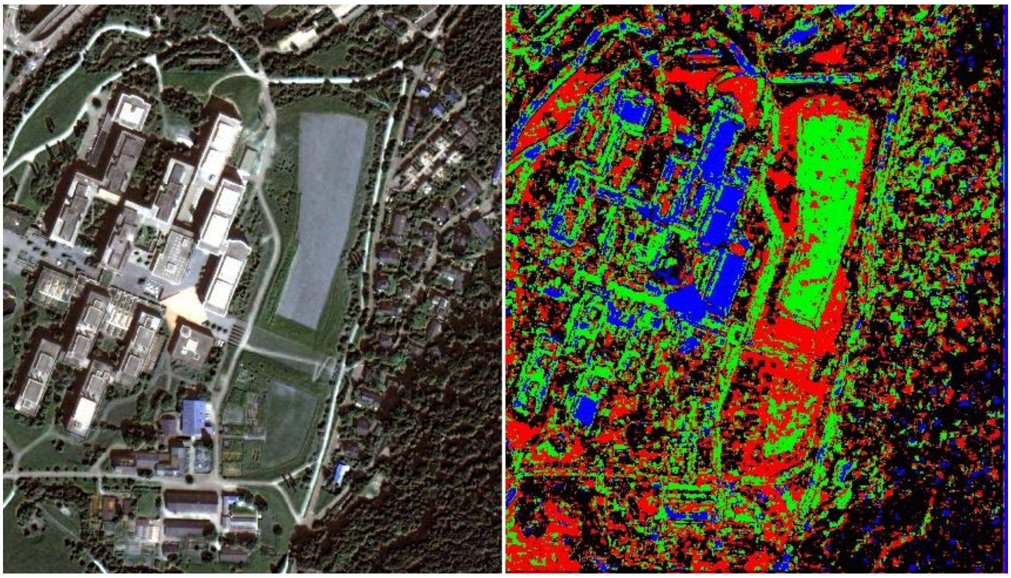
GLCM_2.npy: GLCM calculated for 3 directions and 3 distances and window of 5 for image 2.jpg.

Condition 1:

Image: 2.png
GLCM used: GLCM_2.npy
Number of pixels: 207580
Blurred: yes
Number of clusters: 4
Iterations of kmeans: 10
Epsilon of kmeans: 1.0
Number of Samples per cluster: 2000
RF accuracy: 0.578



Original vs Segmented Image



Original vs RF Classified Image

Input 2:

GLCM_2.npy: GLCM calculated for 3 directions and 3 distances and window of 5 for image 2.jpg.

Condition 2:

Image: 2.png

GLCM used: GLCM_2.npy

Number of pixels: 207580

Blurred: yes

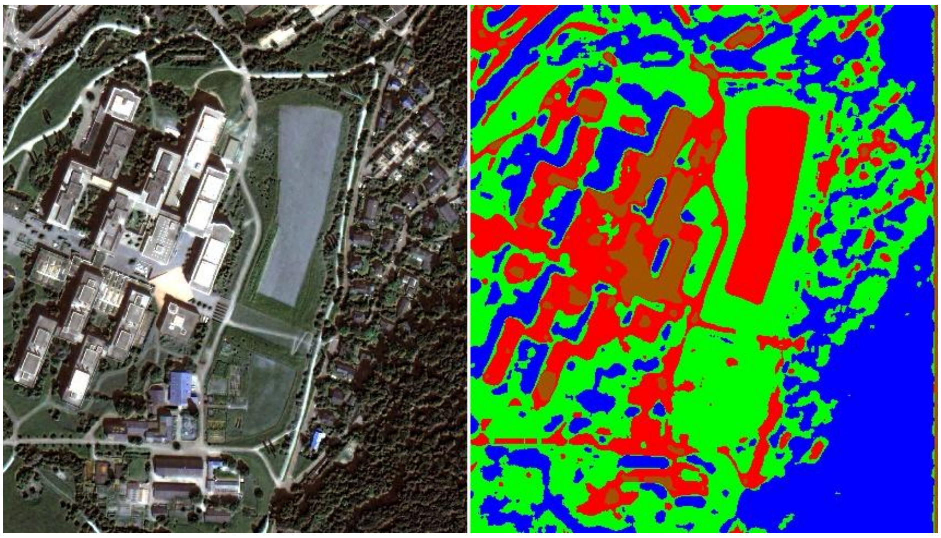
Number of clusters: 4

Iterations of kmeans: 20

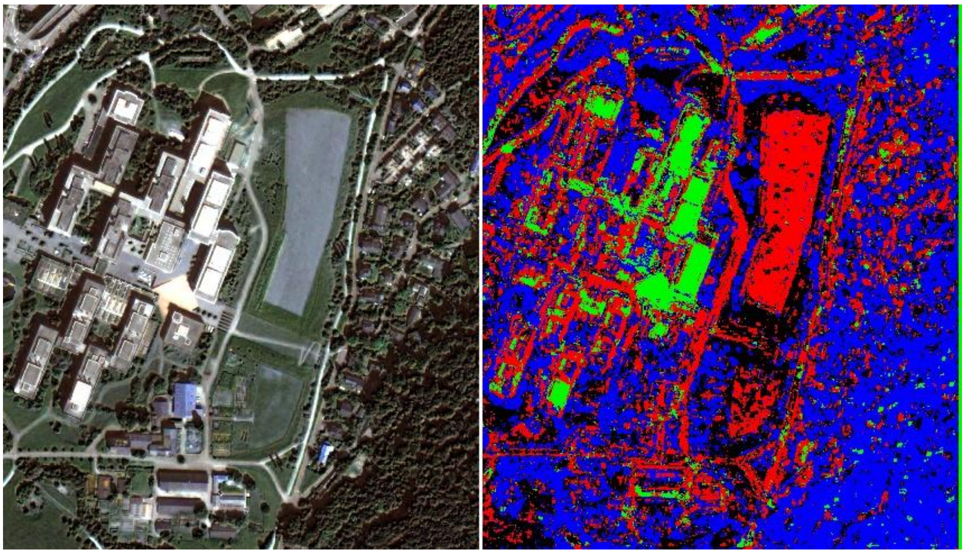
Epsilon of kmeans: 0.1

Number of Samples per cluster: 5000

RF accuracy: 0.606



Original vs Segmented Image



Original vs RF Classified Image

Input 3:

GLCM_3.npy: GLCM calculated for 3 directions and 3 distances and window of 5 for image 3.jpg.

Condition 1:

Image: 3.jpg

GLCM used: GLCM_3.npy

Number of pixels: 22201

Blurred: yes

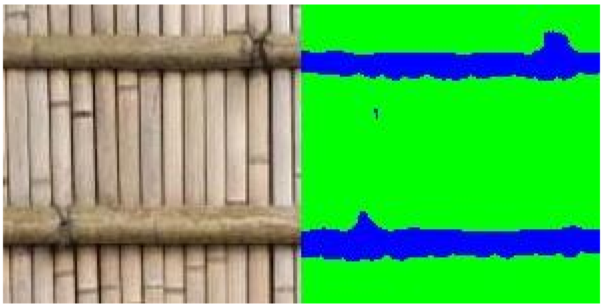
Number of clusters: 2

Iterations of kmeans: 10

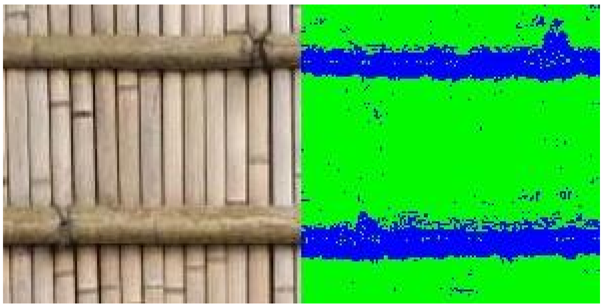
Epsilon of kmeans: 0.1

Number of Samples per cluster: 10000

RF accuracy: 0.9606



Original vs Segmented Image



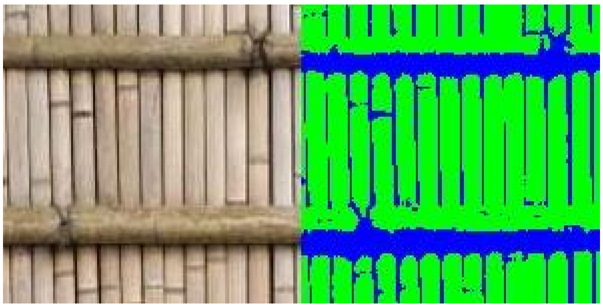
Original vs RF Classified Image

Input 3:

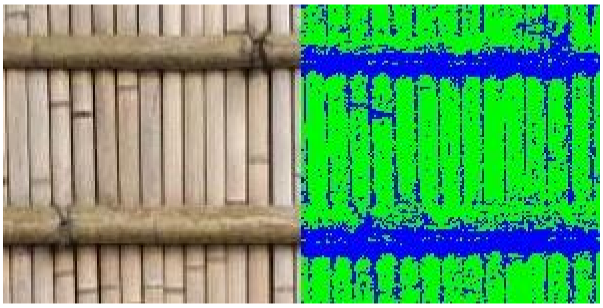
GLCM_3.npy: GLCM calculated for 3 directions and 3 distances and window of 5 for image 3.jpg.

Condition 2:

Image: 3.jpg
GLCM used: GLCM_3.npy
Number of pixels: 22201
Blurred: no
Number of clusters: 2
Iterations of kmeans: 10
Epsilon of kmeans: 0.1
Number of Samples per cluster: 10000
RF accuracy: 0.8968



Original vs Segmented Image



Original vs RF Classified Image

Input 3:

GLCM_3.npy: GLCM calculated for 3 directions and 3 distances and window of 5 for image 3.jpg.

Condition 3:

Image: 3.jpg

GLCM used: GLCM_3.npy

Number of pixels: 22201

Blurred: yes

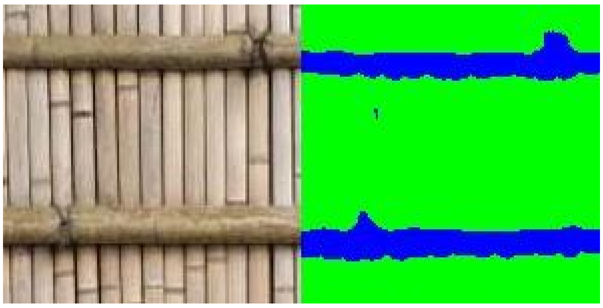
Number of clusters: 2

Iterations of kmeans: 10

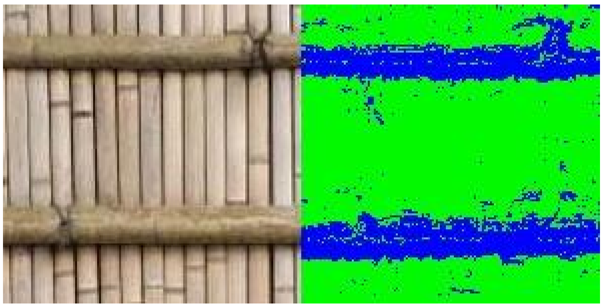
Epsilon of kmeans: 0.1

Number of Samples per cluster: 5000

RF accuracy: 0.9196



Original vs Segmented Image



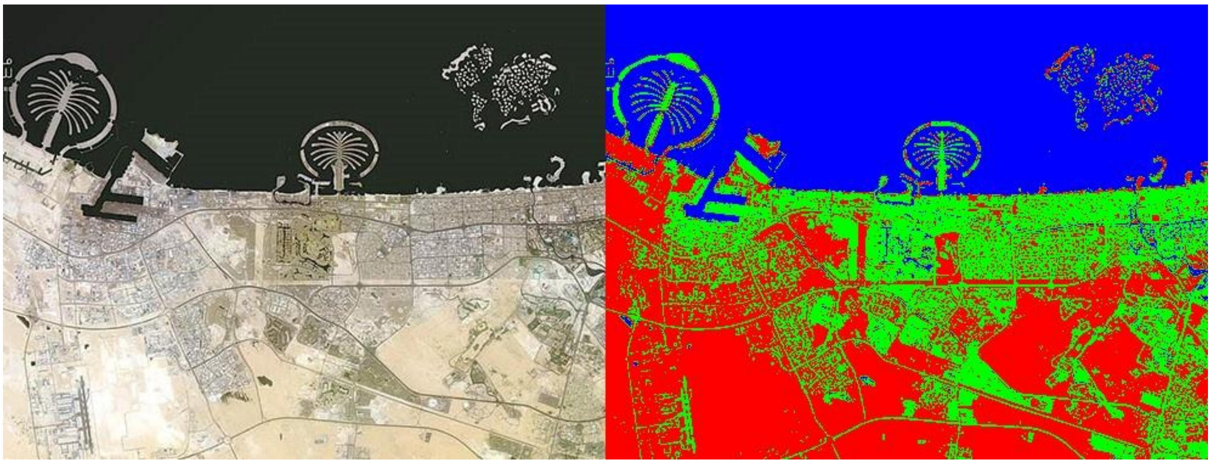
Original vs RF Classified Image

Input 4:

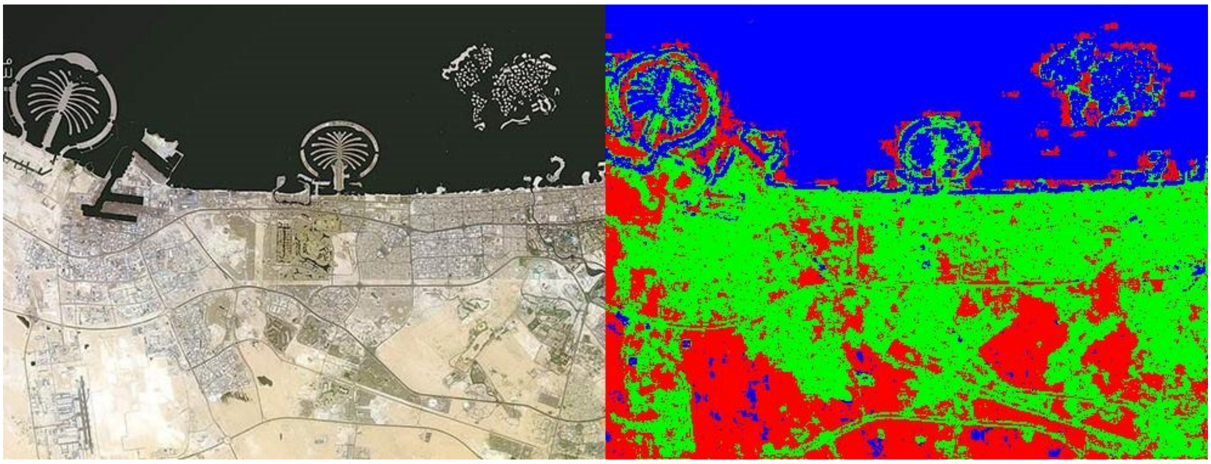
GLCM_4.npy: GLCM calculated for 3 directions and 3 distances and window of 5 for image 4.jpg.

Condition 1:

Image: 4.jpg
GLCM used: GLCM_4.npy
Number of pixels: 185504
Blurred: no
Number of clusters: 3
Max Iterations of kmeans: 10
Epsilon of kmeans: 0.1
Number of Samples per cluster: 3000
RF accuracy: 0.7488888888888889



Original vs Segmented Image



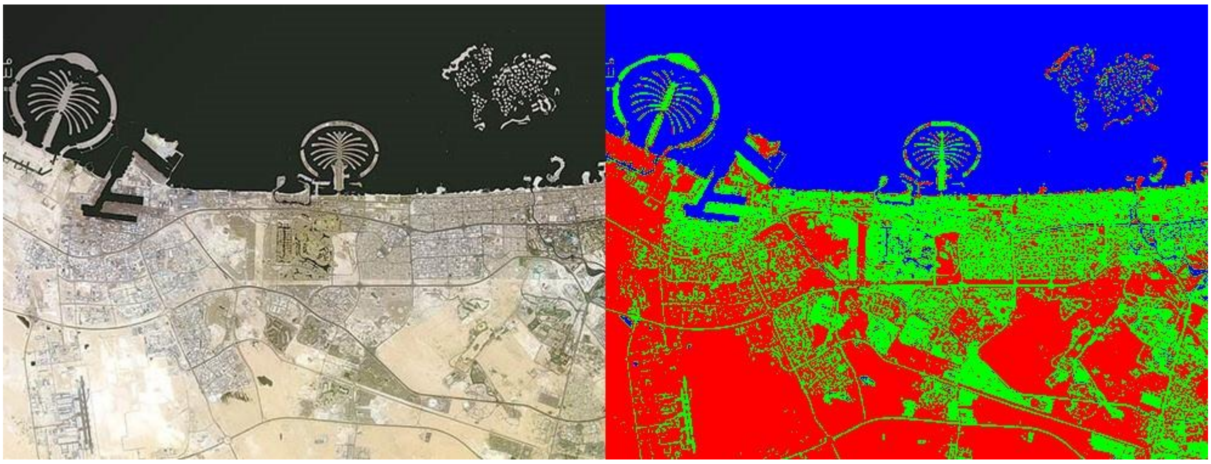
Original vs RF Classified Image

Input 4:

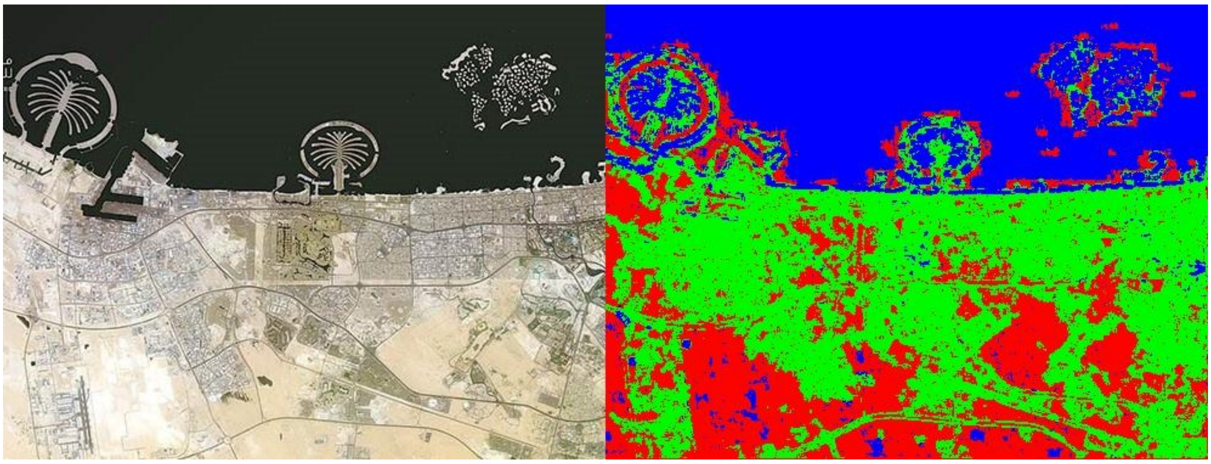
GLCM_4.npy: GLCM calculated for 3 directions and 3 distances and window of 5 for image 4.jpg.

Condition 2:

Image: 4.jpg
GLCM used: GLCM_4.npy
Number of pixels: 185504
Blurred: no
Number of clusters: 3
Max Iterations of kmeans: 10
Epsilon of kmeans: 0.1
Number of Samples per cluster: 3000
RF accuracy: 0.7186666666666667



Original vs Segmented Image



Original vs RF Classified Image

Input 4:

GLCM_4.npy: GLCM calculated for 3 directions and 3 distances and window of 5 for image 4.jpg.

Condition 3:

Image: 4.jpg

GLCM used: GLCM_4.npy

Number of pixels: 185504

Blurred: yes

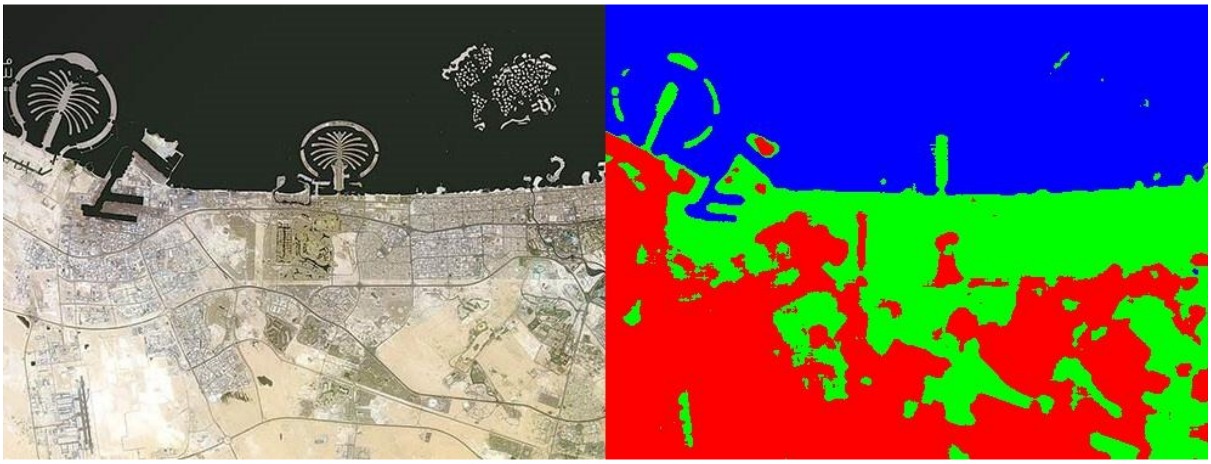
Number of clusters: 3

Max Iterations of kmeans: 10

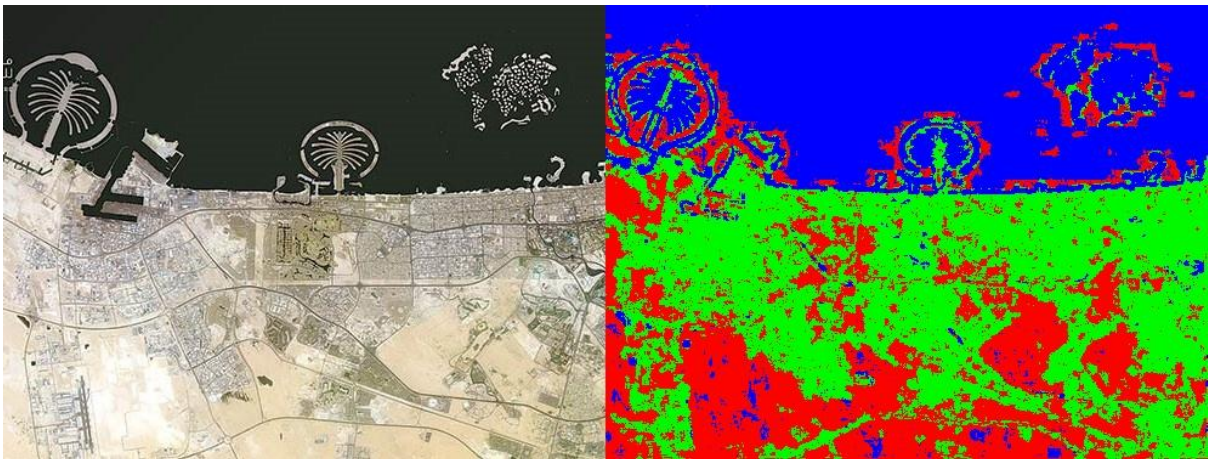
Epsilon of kmeans: 0.1

Number of Samples per cluster: 3000

RF accuracy: 0.7404444444444445



Original vs Segmented Image



Original vs RF Classified Image