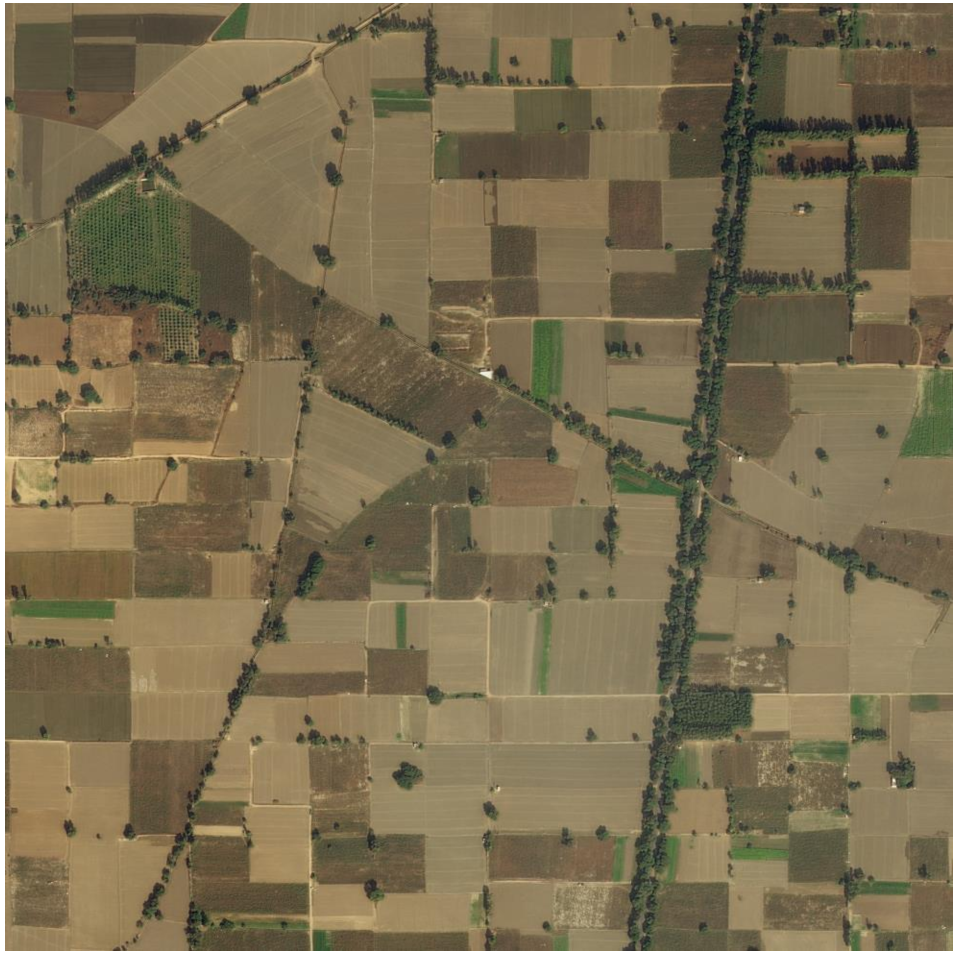
Cluster classification using Laws features of a satellite image

- Shiv Kiran Bagathi 200050019
- Desai Divyeswar reddy 200050033
- Komma Sharanya 210050086

Problem Statement:

- Segment an image using Kmeans clustering then select samples and extract Laws’ texture feature for classification.
- Value of K to be specified by user.
- SVM/RF/ANN can be used as a classifier.

Input Image:



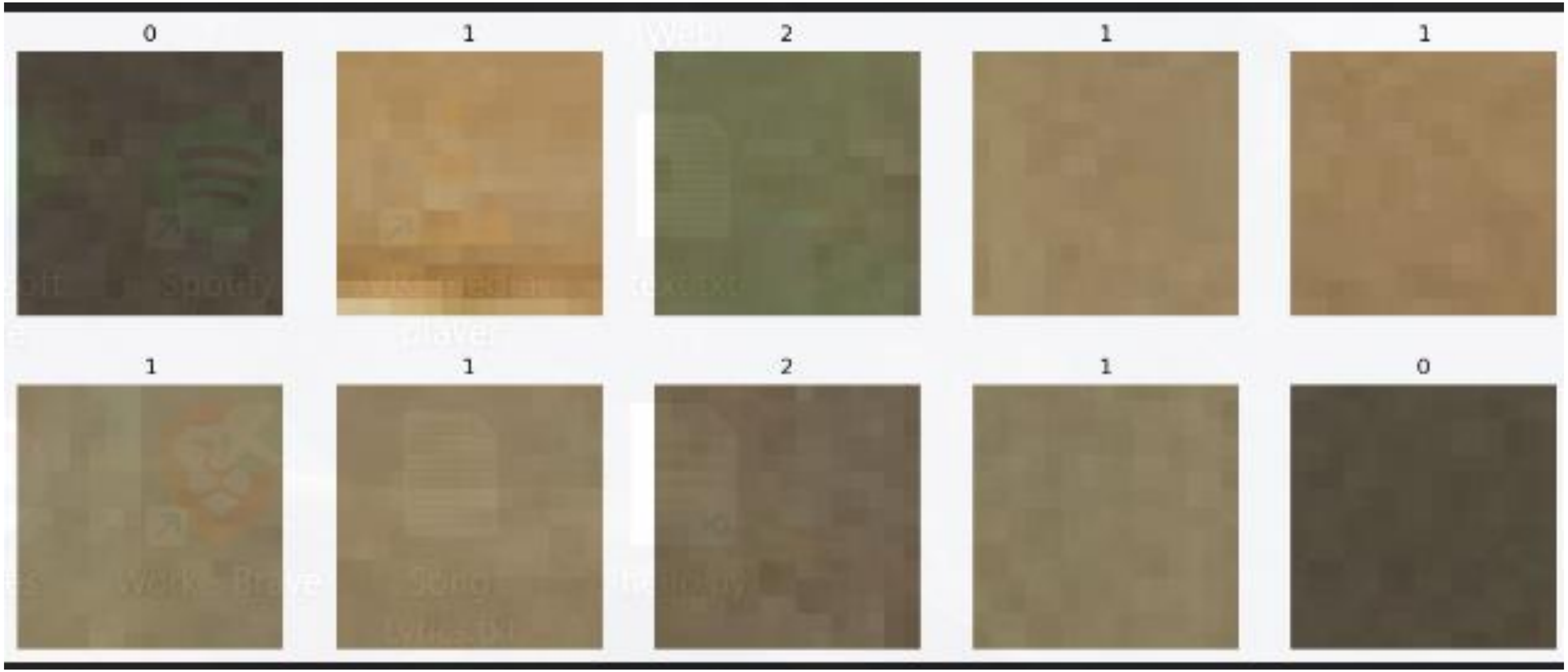
Data and its preprocessing

- The input data consists of a single satellite image which is used for training of the classifier.
- At first we took this image to be a high resolution image but as result we faced the below problems
  1. Since it is a high resolution image applying the kmeans algorithm results in a very distributed labeling of the image and as a result the same labelled clusters are distributed in the image.
  2. While classifying the image since we have to take a pixel into consideration more than once and hence takes a lot of time.
- Hence we opted to use an image with a bit low resolution since a high resolution image is not best for the kmeans approach.
- First, we apply kmeans for clustering this image with given k value(obtained from the GUI by the user).

Data and its preprocessing

- With the obtained values a dataset is created which consists of sample data from the clustered image.
- We sampled random WxW samples whose labels of all pixels are of same cluster
- These samples are obtained from the clusters in the image by using a sliding window method.
- We considered samples of equal number from each cluster.
- But here there is a case of overfitting of lower clustered data.
- Feature energies are extracted from this sample image data using the Laws masks.

Clusters of input image for k =3.



Laws' Texture Features

- Laws Texture description uses masks, which extract the following features
    1. Average Gray Level
    2. Edges
    3. Spots
    4. Ripples
    5. Waves

Masks used to create 5x5 2D Filter Masks

- L5 = [1 4 6 4 1] # level 0
- E5 = [-1 -2 0 2 1] # Edge
- S5 = [-1 0 2 0 -1] # Spot
- W5 = [-1 2 0 -2 1] # Wave
- R5 = [1 -4 6 -4 1] # Ripple
- Using these 5 1D masks we created 25 2D masks to convolve with the input sample to extract out energies

Steps in Feature Extraction

The sampled WxW images are transformed using Law's Features

Convolve sampled pixels with Laws 2D Masks

Non Linearly transformed Laws' filter outputs using sigmoid function

Generated Texture energies by taking sum of squares of Transformed output.

Classifier

- We are using an svm classfier with input shape as "(number of samples,25)" where these 25 vales are the texture energy values.
- We also tried this using Random forest classifier but since svm is showing the best results among these two opted for svm.
- This feature matrix is then passed into the svm classifer with a rbf kernel .
- Regarding the kernel we tried with rbf poly and linear and the order of accuracy in the cases is RBF>Polynomial>Linear.
- This trained model is then tested with the test data.

Training the SVM classifier:

- Generated Energies features are trained using SVM classifier with regularizing constant C =1.
- Input dimension of classifier is (k*each_cluster_data, 20)
- Output prediction of the classifier is (k*each_cluster_data)
- Ouput values range from 0 to k-1.
- From the original image, We considered window of wxw and assigned the cluster value of the sub-image to the whole wxw pixels

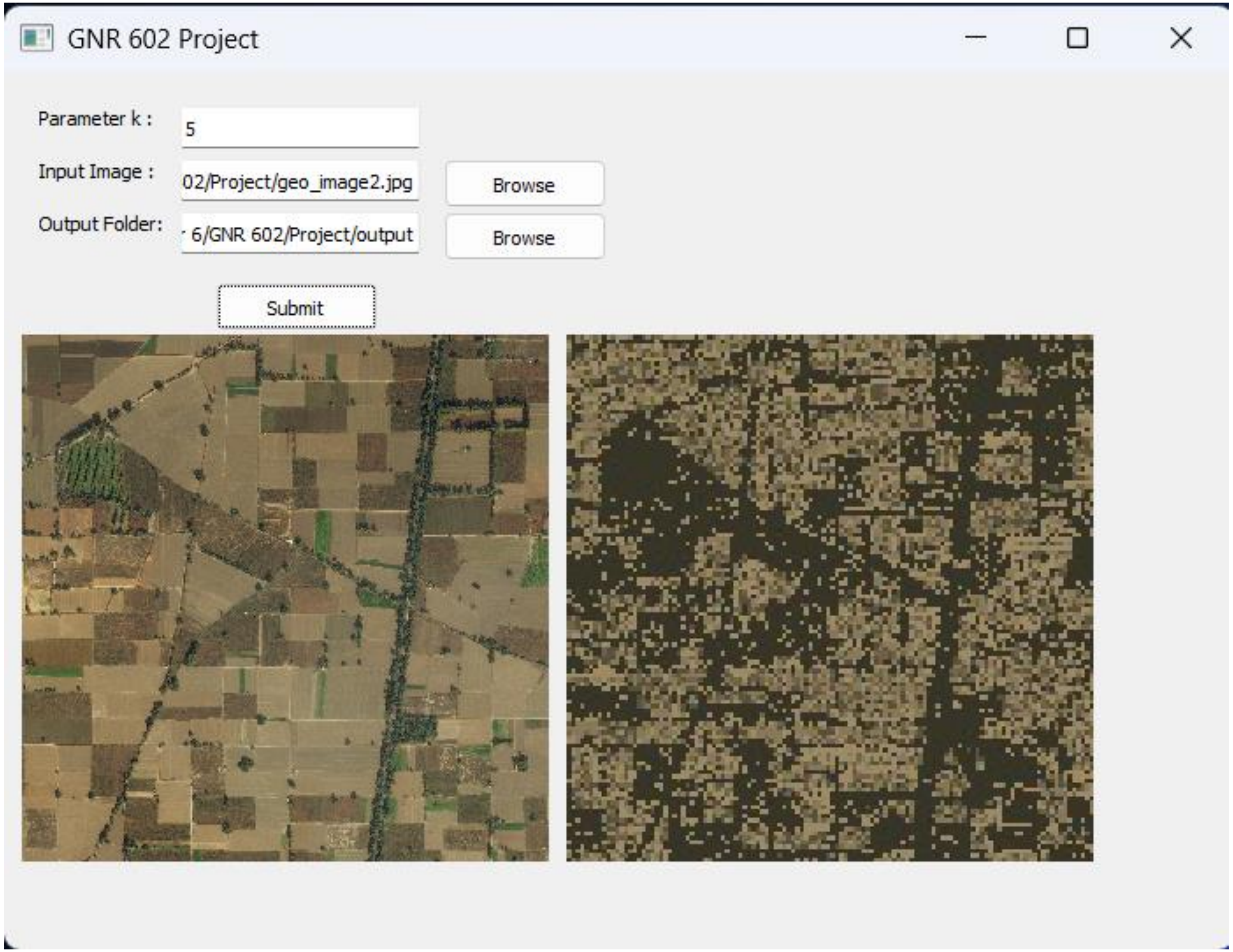
Using the Trained Classifier

- Using the trained classifier, we now try to cluster our input image.
- For this process, We segmented the image into equal wxw blocks.
- Now, the label of this window will be assigned to whole wxw pixels.
- So final clustered image is of resolution (original_x/w, original_y/w)
- Finally, we output the segmented image into the required folder as out.jpg file.

Some Observations

- While clustering, Kmeans took in considering of color as a parameter, while using texture features we don't consider color as a parameter.
- Also, while training taking sample amount from different clustering can induce noise in the border regions of textures. So when we segment the image into equal windows, there is significant error at edge region of texture.
- These are reason for some error in re-clustering using classification

GUI

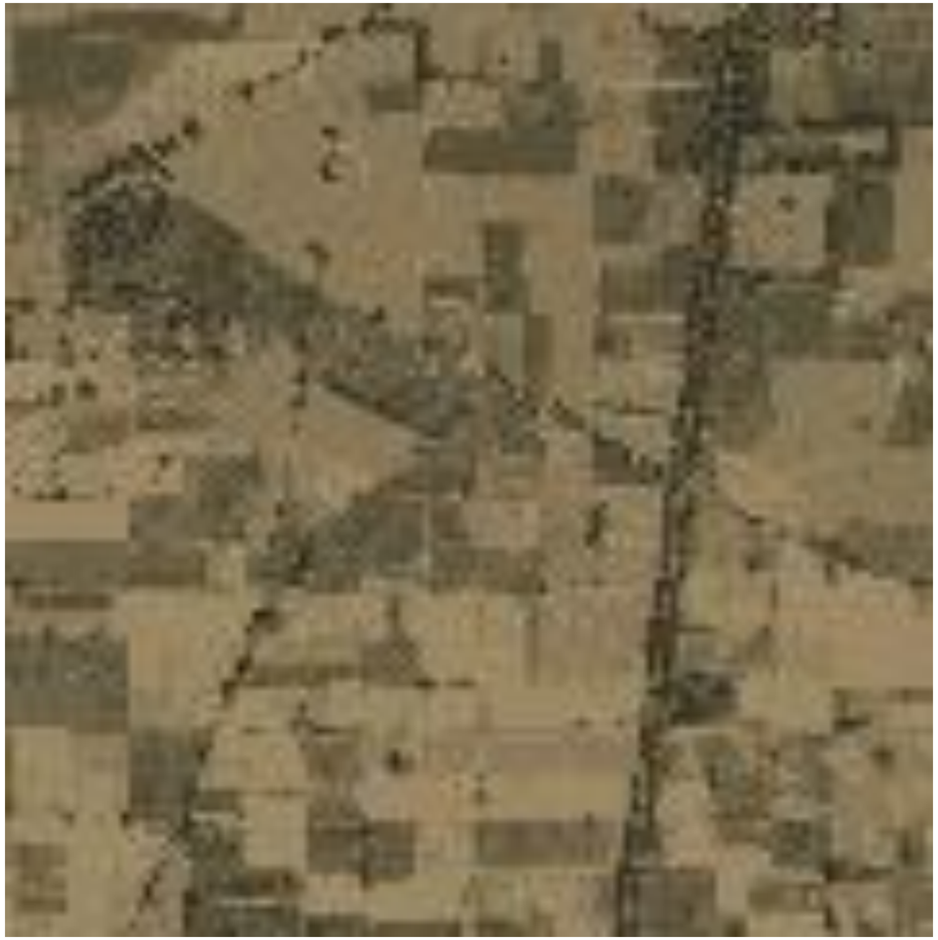


Parameter k : 5

Input Image : 02/Project/geo_image2.jpg

Output Folder: · 6/GNR 602/Project/output

Clustered Image from Kmeans and Trained Classifier



Thank You