,

FML Project Report

Ajay Ravindran - 213059004
Amitesh Gupta - 214310001
Andrea Austin Furtado - 213056001
Khajjayam Gowtham - 213050021
Paramveer Choudhary - 213050061

27th December, 2021

1 Problem Statement and background

Mapping and monitoring of land use land cover (LULC) is old but still a task of golden value because of its numerous applications in different administrative or business sectors. With the use of LANDSAT series of satellite data, this classification has become a very common practice since the last two three decades, however there is still various chances of further improvements as it deal with several limitations. Other than the acquisition techniques, the classification algorithms and methods have been excelled extensively in last few years which helped to achieve very significant accuracy and precision of classification. Unlike LANDSAT, the landuse landcover classification using Sentinel-2 data is an upcoming trend and becoming vigorously popular because of its higher resolution and accuracy. Meanwhile, the development of different classification algorithms are also mention worthy. Earlier, ISODATA, Minimum distance, parallelepiped and maximum likelihood classifiers were mostly used, whereas nowadays various clustering techniques like k-mean clustering and image segmentation are becoming quite applicable as well as machine learning applications like Support Vector Machines (SVM) and Neural Networks (NN) are also found very effective and efficient to produce excellent outcomes in terms of landuse maps. In the present study, we have incorporated k-mean clustering, image segmentation, SVM and NN approaches to exhibit its potential and effectiveness of image classification.

2 Dataset

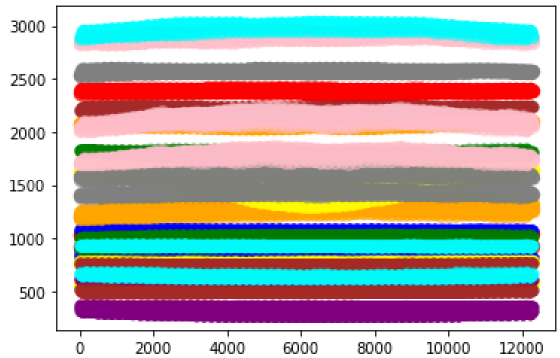
Sentinel-2A, 2B are the two identical satellites which carry optical sensor with 13 bands in Visible and Near Infrared region. Other than Red, Green and Blue, it has one coastal aerosol band, four red-edge bands dedicated for the vegetation condition assessment, one Near Infrared, two Short-Wave Infrared, one water vapour and cirrus cloud absorption bands. Both the satellites have 10 meters of spatial resolution for optical bands and the remaining bands have resolution of 20 meters, the swath has 270km width and repetitive pass or temporal resolution of 10 days. Thus one can able to monitor any land feature once in every five days. In the present study, sentinel-2A data sets have been acquired from GitHub repository. Among these 13 bands, we selected 3 bands - band number 3 (green), 4(red) and 6(red-edge) for forming False Color Composite (FCC) combination so that the task of landuse landcover classification can be easier because using FCC the land features become much distinctive.

3 Classification using K-means clustering

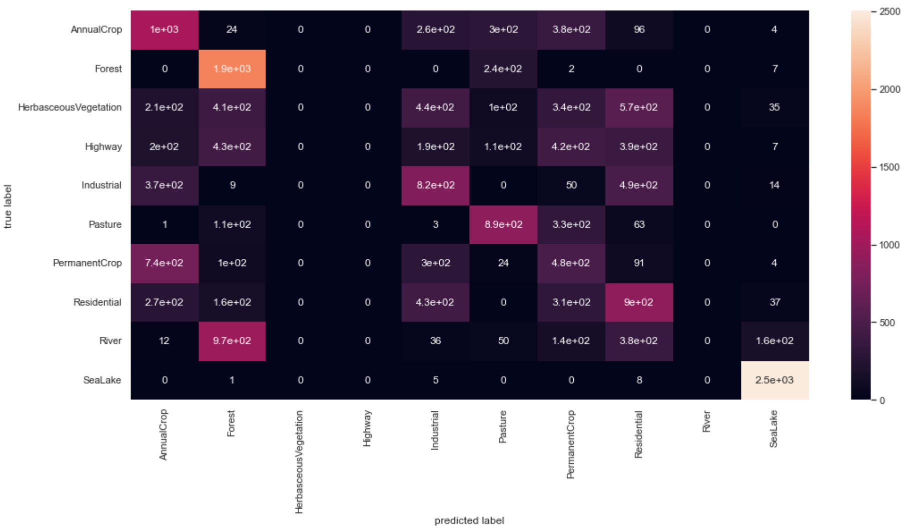
- 19317 tif images from sentinel-2 EuroSat dataset were used as input to k-means clustering. Images in the dataset were labelled and belonged to one of the following 10 classes: Annual Crop, Forest, Herbaceous Vegetation, Highway, Industrial, Pasture, Permanent Crop, Residential, River or Sea Lake.
- Out of the 13 bands that each image had, the 3 most important bands were used for training the model, namely VNIR, Red and Green bands.
- The images were resized to 64X64 size with 3 bands.
- The images were then normalised by dividing vector by 255.
- As a final pre-processing step, the image vectors were flattened.

3.1 Results using K-Means Clustering

- The accuracy obtained from K-Means clustering with number of clusters as 10 (as the data too had 10 classes) was 0.44.
- The following plot shows the scatter plot of vectorised form of the images as obtained after K-Means clustering with number of clusters as 10. It can be seen that there is significant overlap between the clusters.



- The following is the heat-map of the ten classes classes as obtained after clustering. As can be seen Forest and Sea Lake are the best clustered classes.

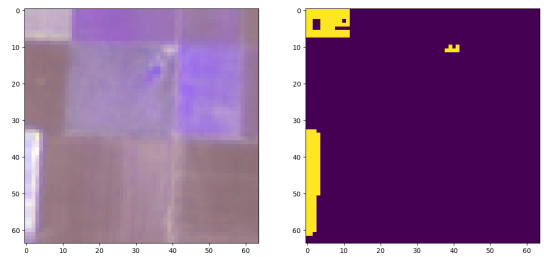


4 Classification using segmentation

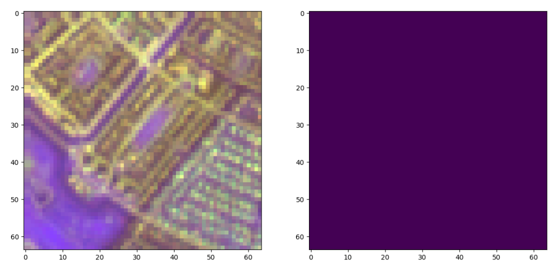
- Here, we are using a clustering technique called mean shift clustering.
- This method doesn't require us to give no of clusters before hand.
- This performs pixel-wise labeling.

4.1 Results using segmentation

The below is image of agricultural land area. It perfectly classifies the agricultural land area and non-agricultural land area.



The below is the image of city area and due it high heterogeneity it is classified into a single class.



5 Classification using SVM

- 27000 tif images were used as dataset. Images in the dataset were labelled and belonged to one of the following 10 classes: Annual Crop, Forest, Herbaceous Vegetation, Highway, Industrial, Pasture, Permanent Crop, Residential, River or Sea Lake.
- Out of the 13 bands that each image had, the 3 most important bands were used for training the model, namely VNIR, Red and Green bands.
- After filtering out the other bands, the image was resized to a 50x50 image and flattened so as to make training faster.

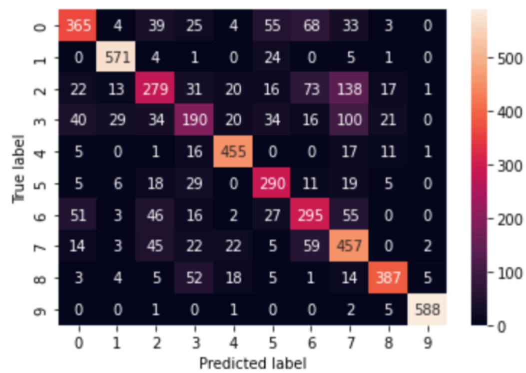
5.1 Results using various kernels

Linear, polynomial and RBF kernels were tried for classification. RBF kernel was seen to give the highest accuracy, followed by polynomial kernel and linear kernel.

5.1.1 RBF kernel

An accuracy of 72% was obtained when RBF kernel was used. The precision was observed to be 0.71 and the recall was 0.72.

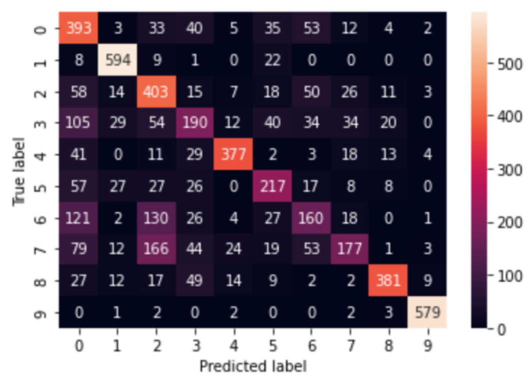
	precision	recall	f1-score	support
0	0.72	0.61	0.66	596
1	0.90	0.94	0.92	606
2	0.59	0.46	0.52	610
3	0.50	0.39	0.44	484
4	0.84	0.90	0.87	506
5	0.64	0.76	0.69	383
6	0.56	0.60	0.58	495
7	0.54	0.73	0.62	629
8	0.86	0.78	0.82	494
9	0.98	0.98	0.98	597
accuracy			0.72	5400
macro avg	0.71	0.72	0.71	5400
weighted avg	0.72	0.72	0.71	5400



5.1.2 Polynomial kernel

An accuracy of 64% was obtained when polynomial kernel was used. The precision was observed to be 0.65 and the recall was 0.63.

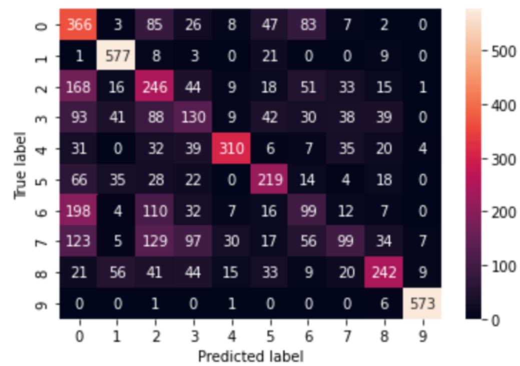
	precision	recall	f1-score	support
0	0.44	0.68	0.54	580
1	0.86	0.94	0.89	634
2	0.47	0.67	0.55	605
3	0.45	0.37	0.41	518
4	0.85	0.76	0.80	498
5	0.56	0.56	0.56	387
6	0.43	0.33	0.37	489
7	0.60	0.31	0.40	578
8	0.86	0.73	0.79	522
9	0.96	0.98	0.97	589
accuracy			0.64	5400
macro avg	0.65	0.63	0.63	5400
weighted avg	0.65	0.64	0.64	5400



5.1.3 Linear kernel

An accuracy of 53% was obtained when linear kernel was used. The precision was observed to be 0.53 and the recall was 0.52.

	precision	recall	f1-score	support
0	0.34	0.58	0.43	627
1	0.78	0.93	0.85	619
2	0.32	0.41	0.36	601
3	0.30	0.25	0.27	510
4	0.80	0.64	0.71	484
5	0.52	0.54	0.53	406
6	0.28	0.20	0.24	485
7	0.40	0.17	0.23	597
8	0.62	0.49	0.55	490
9	0.96	0.99	0.98	581
accuracy			0.53	5400
macro avg	0.53	0.52	0.52	5400
weighted avg	0.53	0.53	0.52	5400



5.2 Analysis of results

- Good results were obtained in classifying images belonging to Forest and Sea Lake classes. The homogeneity of the images in these classes made them easier to be classified.
- The classifiers had a tough time classifying images belonging to the classes Highway, Permanent Crop and Residential. The images in these classes were more difficult to classify due to their heterogeneous nature.

6 Multi-label Land Cover Classification using fastai

- UCMerced Dataset was used. It contains 2100 multi-labelled images belonging to 17 classes.
- The loss function used is Flattenedloss of BCEwithLogitsLoss.
- The framework used is fastai and Resnet34 model was employed.
- An accuracy of **95.03%** was obtained.

6.1 Approach used

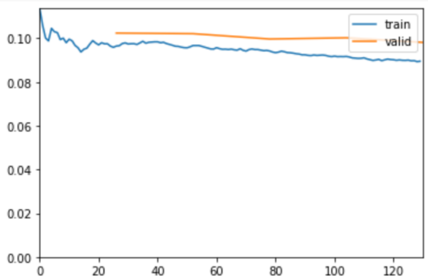
- The image size used for training is 256x256. Images were also transformed using get_transforms for data augmentation.
- Resnet34 trained on Imagenet is used for learning data.
- fit_one_cycle() method, is used due to its better performance in speed and accuracy, over the fit() method. fit_one_cycle() is Fastai's implementation of Leslie Smith's 1cycle policy - "The essence of this learning rate policy comes from the observation that increasing the learning rate might have a short term negative effect and yet achieve a longer term beneficial effect."

6.2 Associated papers

- Cyclical Learning Rates for Training Neural Networks. (2017)
- Super-Convergence: Very Fast Training of Neural Networks Using Large Learning Rates. (2018)
- A disciplined approach to neural network hyper-parameters: Part 1 — learning rate, batch size, momentum, and weight decay. (2018)

6.3 Results

epoch	train_loss	valid_loss	accuracy_thresh	fbeta	time
0	0.095851	0.102366	0.954482	0.910554	00:17
1	0.096068	0.102089	0.953361	0.908405	00:17
2	0.094426	0.099658	0.955882	0.912108	00:18
3	0.091596	0.100200	0.956443	0.915057	00:18
4	0.089498	0.098099	0.957563	0.913911	00:17



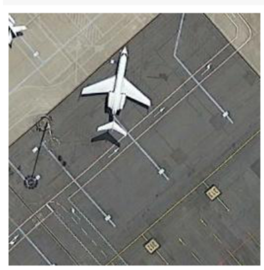

```
Better model found at epoch 0 with valid_loss value: 0.10236554592847824.
Better model found at epoch 1 with valid_loss value: 0.10208877176046371.
Better model found at epoch 2 with valid_loss value: 0.09965842962265015.
Better model found at epoch 4 with valid_loss value: 0.09809941798448563.
```

```
img = open_image("/content/test/roundabout_086.jpg")
img
```

```
pred_class,pred_idx,outputs = learn.predict(img)
print(pred_class)
```

```
bare-soil;buildings;cars;grass;pavement
```

```
img2 = open_image("/content/test/airplane_439.jpg")
img2
```



```
pred_class,pred_idx,outputs = learn.predict(img2)
print(pred_class)
```

```
airplane;pavement
```

7 Land cover segmentation using Keras

- Dubai aerial imagery dataset was used. It contains 72 images which belonged to 6 classes
- The framework used is keras and Resnet34 model trained with Imagenet was employed.
- An accuracy of **83.4%** was obtained.

7.1 Approach used

- The 72 images in the dataset and their corresponding masks were divided into images of the size 256x256 using patchify. This resulted in a total of 1305 images to train from.
- Each pixel in the mask was converted from RGB to corresponding label class(integer). The mapping of the class and integer is as follows:

Class	**RGB**	**Integer**
Building	3C1098	0
Land	8429F6	1
Road	6EC1E4	2
Vegetation	FEDD3A	3
Water	E2A929	4
Unlabeled	9B9B9B	5

- Unet was used to train on these images.Input to this model is 256x256 image.

7.2 Results

```
Total params: 1,941,190
Trainable params: 1,941,190
Non-trainable params: 0
________________________________________________________________________________________________
Epoch 1/10
209/209 [==============================] - 19s 43ms/step - loss: 0.9832 - accuracy: 0.6401 - jacard_coef: 0.3525 - val_loss: 0.9728 - val_accuracy: 0.6651 - val_jacard_coef: 0.4351
Epoch 2/10
209/209 [==============================] - 8s 36ms/step - loss: 0.9557 - accuracy: 0.7362 - jacard_coef: 0.5019 - val_loss: 0.9659 - val_accuracy: 0.7001 - val_jacard_coef: 0.4896
Epoch 3/10
209/209 [==============================] - 7s 36ms/step - loss: 0.9496 - accuracy: 0.7568 - jacard_coef: 0.5354 - val_loss: 0.9547 - val_accuracy: 0.7408 - val_jacard_coef: 0.5251
Epoch 4/10
209/209 [==============================] - 7s 36ms/step - loss: 0.9428 - accuracy: 0.7805 - jacard_coef: 0.5691 - val_loss: 0.9473 - val_accuracy: 0.7639 - val_jacard_coef: 0.5487
Epoch 5/10
209/209 [==============================] - 7s 36ms/step - loss: 0.9367 - accuracy: 0.7983 - jacard_coef: 0.5976 - val_loss: 0.9411 - val_accuracy: 0.7833 - val_jacard_coef: 0.5790
Epoch 6/10
209/209 [==============================] - 7s 36ms/step - loss: 0.9329 - accuracy: 0.8100 - jacard_coef: 0.6165 - val_loss: 0.9386 - val_accuracy: 0.7900 - val_jacard_coef: 0.5892
Epoch 7/10
209/209 [==============================] - 7s 36ms/step - loss: 0.9301 - accuracy: 0.8156 - jacard_coef: 0.6278 - val_loss: 0.9364 - val_accuracy: 0.8033 - val_jacard_coef: 0.6123
Epoch 8/10
209/209 [==============================] - 7s 36ms/step - loss: 0.9269 - accuracy: 0.8247 - jacard_coef: 0.6431 - val_loss: 0.9330 - val_accuracy: 0.8115 - val_jacard_coef: 0.6182
Epoch 9/10
209/209 [==============================] - 7s 36ms/step - loss: 0.9253 - accuracy: 0.8289 - jacard_coef: 0.6497 - val_loss: 0.9331 - val_accuracy: 0.8153 - val_jacard_coef: 0.6225
Epoch 10/10
209/209 [==============================] - 7s 36ms/step - loss: 0.9233 - accuracy: 0.8341 - jacard_coef: 0.6580 - val_loss: 0.9293 - val_accuracy: 0.8284 - val_jacard_coef: 0.6426
```