IIT PALAKKAD

The Eye in the Sky

7th Inter IIT Tech Meet

Rajendra Singh
Muskaan Maheshwari

Contact Information:
Rajendra Singh
Email id: 111601017@smail.iitpkd.ac.in
Muskaan Maheshwari
Email id: 121701014@smail.iitpkd.ac.in

INTRODUCTION

India needs to be digitally mapped. This was manually done about 80 years ago by the British. Most of the existing services of GPS are provided based on the data extracted and inserted manually on to the databases of online map resources like Google Earth and Wikimapia. The advancement in Image processing is necessary to terminate the heavy expense of imagery analytics and interpretation often performed by human imagery analysts.

Object recognition is a significant challenge in Image processing. Applications of object recognition can be found in the areas of automated navigation, object tracking, video surveillance, environmental monitoring, natural resource management. It has been observed experimentally that the classical image processing algorithms fail when used on intricate natural images like satellite images. At the same time, there is also an abundance of satellite imagery. In addition, the capturing of remote sensor images is not restricted by weather, time, environmental issues. This indicates that a large number of remote sensor images can be produced as per need.

For a satellite image, machine learning creates the view on the ground

Machine learning is a broad area which encompasses various techniques. Each of these techniques is designed and applicable for specific tasks. It remains to be identified what machine learning approach is best suited for object recognition in satellite images. There are several implementations under the classification of Machine Learning i.e. supervised learning and unsupervised learning .

Satellite images hold a large amount of data and can be considered under the classification of complex natural images. Object based image analysis meets the demand of handling large data within a limited time. Object based classification reinforce the analysis performed on satellite images for perception of urban features. It is therefore essential to understand how efficient object recognition can be performed on them by the application of machine learning techniques. To identify a suitable approach for identifying these objects in satellite images is the aim of this report.

Classification Approach

Motivation

Satellite image classification can be done broadly by the following three ways:-

1) Unsupervised learning
2) Supervised learning
3) OBIA(Object based Image Analysis)

We chose to work with Supervised way of learning, because of the following reasons:-

1) Supervised v/s Unsupervised Learning

The classification method depends on the quality of the spectral data in which the classification algorithm is to be used and the level of class detail required.

Unsupervised classification algorithms require the analyst to assign labels and combine classes after the fact into useful information classes (e.g forest, agriculture, water, etc). In many cases, this after the fact assignment of spectral clusters is difficult or not possible because these clusters contain assemblages of mixed land cover types. Generally, It is useful for quickly assigning labels to uncomplicated, broad land cover classes such as water, vegetation/non-vegetation, forested/non-forested, etc. Furthermore, unsupervised classification may reduce analyst bias.

Supervised classification allows the analyst to fine tune the information classes, often to much finer subcategories, such as species level classes. Training data is collected in the field with high accuracy GPS devices or expertly selected on the computer. Consider for example if you wished to classify percent crop damage in corn fields. A supervised approach would be highly suited to this type of problem because you could directly measure the percent damage in the field and use these data to train the classification algorithm. Using training data on the result of an unsupervised classification would likely yield more error because the spectral classes would contain more mixed pixels than the supervised approach.

2) Supervised v/s OBIA

We used **deep convolutional neural network** (supervised method) to solve this problem. An important advantage of the **CNN-based** detector is that it required less human supervision than OBIA, can be trained using a relatively small number of samples, and can be easily transferable to other regions or scenes with different characteristics, e.g., color, extent, light, background, or size and shape of the target objects. The lack of direct transferability was an important limitation of OBIA methods. Since, once calibrated for one image, the OBIA settings are not directly portable to other images (e.g., to different areas, extensions, radiometric calibrations, background color, spatial and spectral resolutions, or different sizes or shapes of the target objects).

Methodology

1) Creating Small Patches:

The Satellite images were less in number, but of very high resolution. So, we chose to divide the image in patches of size 4x4x4. Later on, we will use these compact patches to train our Network. We will create final output images by joining the small patches predicted by the model.

2) Batch Normalization:

We used **batch normalisation** as it allows much higher learning rates, increasing the speed at which networks train. It makes weights easier to initialise as weight initialisation can be difficult, especially when creating deeper networks. **Batch normalisation** helps reduce the sensitivity to the initial starting weights

3) Regularization method:

We used **Dropout** technique to reduce **overfitting** while training neural networks to prevent complex co-adaptations on training data. The role of hidden units in neural networks is to approximate a 'function' efficiently from the available data-samples which can be generalized to unseen data. It is a very efficient way of performing model averaging with neural networks. The term "**dropout**" refers to **dropping out** units (both hidden and visible) in a neural network.

4) Callbacks:

We used callbacks as they can help in fixing bugs more quickly, and can help in building better models. It becomes easier to visualise the model’s training, and can prevent overfitting by customizing the learning rate on each iteration.

5) Cross-validation:

Cross-validation, sometimes called rotation estimation, is a model validation technique for assessing how the results of a statistical analysis will generalize to an independent data set.

It is mainly used in settings where the goal is prediction, and one wants to estimate how accurately a predictive model will perform in practice.
We also did k-fold cross validation and so as to prevent model from biasing toward any particular validation set.

Later, we ensemble this model to create more generalised model and predicted output.

6) Data Augmentation:
As no. of image is less we could perform data augmentation to increase the number of training images.

Implementation

1) **Generating dataset for training** :
Given a four band image, we created a patch of size 4x4x4. Array of such patches is passed through the model as input.
2) **Creating Labels for model**:
We processed the given labels for the train images into patches of size same as **patchsize.** The colours occurring maximum number of times in the label's patch was used to generate a One-Hot encode vector for Model training.
3) **Model Training**:
3.1)Architecture of CNN model:

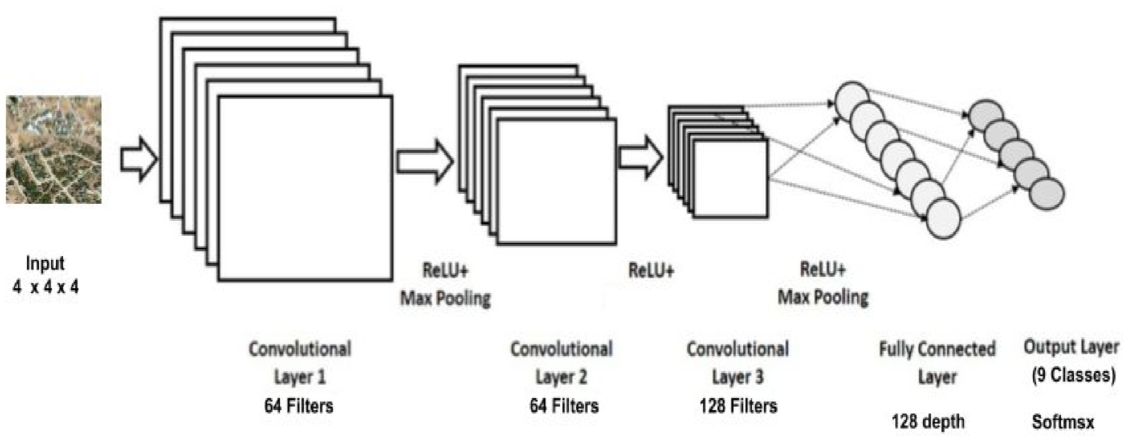


Fig3.1: CNN Architecture with input size as (4,4,4)

Below is the **model summary** on which the images has been trained:

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 4, 4, 64)	320
Batch_normalization_1	(Batch (None, 4, 4, 64)	256
activation_1 (Activation)	(None, 4, 4, 64)	0
max_pooling2d_1 (MaxPooling2)	(None, 2, 2, 64)	0
conv2d_2 (Conv2D)	(None, 2, 2, 64)	4160
Batch_normalization_2	(Batch (None, 2, 2, 64)	256
activation_2 (Activation)	(None, 2, 2, 64)	0
conv2d_3 (Conv2D)	(None, 2, 2, 128)	73856
batch_normalization_3	(Batch (None, 2, 2, 128)	512
activation_3 (Activation)	(None, 2, 2, 128)	0
max_pooling2d_2 (MaxPooling2)	(None, 1, 1, 128)	0
flatten_1 (Flatten)	(None, 128)	0
output_layer (Dense)	(None, 9)	1161

Total params:	80,521
Trainable params:	80,009
Non-trainable params:	512

3.2) Parameters Selection:

3.2.1) We take **Patchsize** = 4 (i.e. 4x4x4 as Input), as to creates more training data while keeping the surrounding info of pixel intact. Too big patchsize will lead to lesser number of training data and also with very small patchsize local info will be lost, both is unfavorable.

3.2.2) Batchsize = 128

3.2.3) Epoch = 50

3.3) We chose '**categorical_crossentropy**' as loss for our keras model.

3.4) Optimizer Selection :

We chose **Nadam** (Nesterov Adam Optimizer)to train the model since it is stochastic optimization method, which trains quickly on big data. Also, Nadam is Adam RMSprop with Nesterov momentum.

3.5) Dynamic Learning Rate:

We used keras callback **ReduceLROnPlateau** as it reduces the learning rate by a factor of 2-10, once learning stagnates. This callback monitors a quantity and if no improvement is seen for a **patience** number of epochs, the learning rate is reduced.

3.6) Checkpoints:

We used **ModelCheckpoint** from Keras Model to save the latest best model according to the quantity monitored.

3.7) EarlyStopping:

We used **EarlyStopping** from Keras Model to save time as training will stop if the model doesn't show improvement over the accuracy of validation set and restore model weights from the epoch with the best value of the monitored quantity.

4) **Model Prediction**

4.1) White Classification:

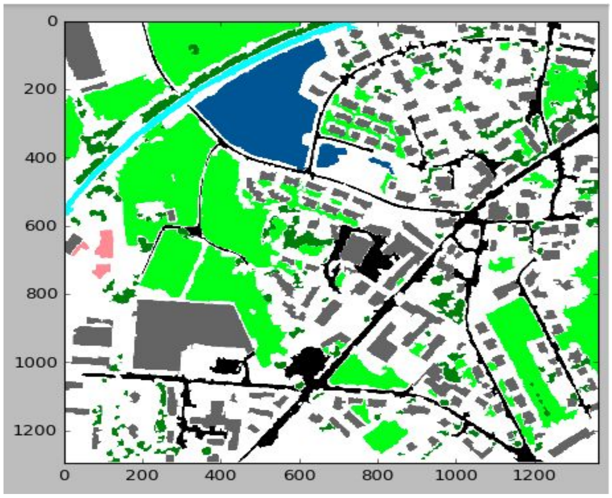
As we know that white colour represents "NO CLASS" label i.e. the area represented by the pixels is not known. By increasing the **Alphawhite** variable in the code, we can decrease the False-positive of No Class (White).

4.2) Output processing:

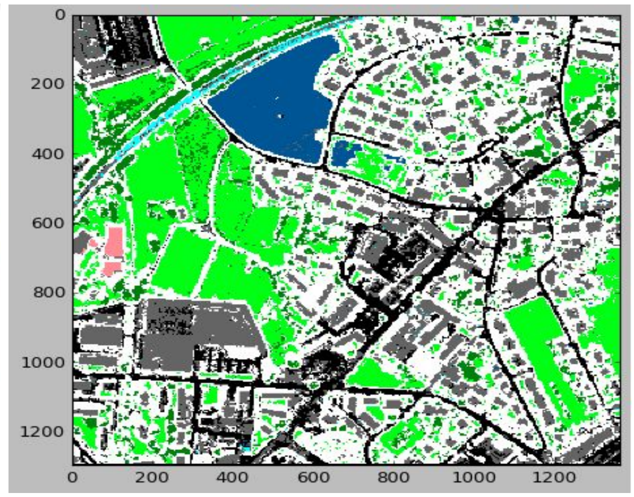
For each input patch, model will produce an output consisting of probabilities associated with the class it belongs to. We take argmax of that output vector to get a class with maximum probability. Then, we create a new image of size same as patchsize with each entry as the predicted class (i.e color associated with that class). We join all such new images to create a final output image (size same as the labelled image).

5) **Smoothing**:

For better visualization, we smoothen the predicted image from our model by running max pooling with (2,2) and strides in {1,2,3}.



(a)



(b)

(c)

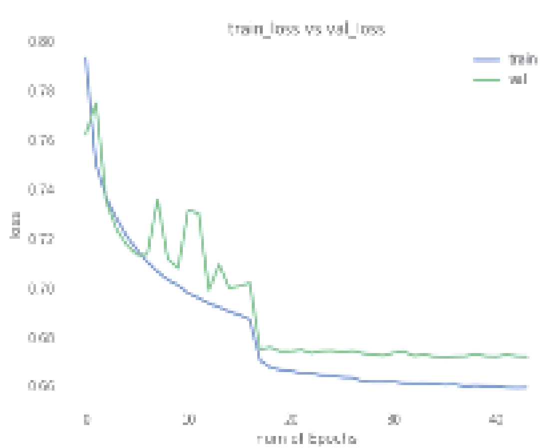
Fig 5.1(a)Training Image, (b)Predicted Image(Before Smoothening), (c)Smoothened Image

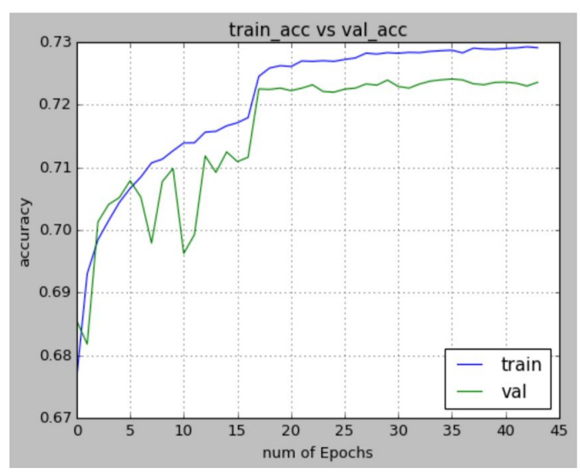
Results

While training, the white colour is been taken in consideration. The Classification Map for the 9 classes is as follows:

ID	Pixel Value	Colour	Class
0	(255, 255, 255)	White	NO CLASS
1	(100, 100, 100)	Gray	Building
2	(0, 0, 0)	Black	Road
3	(0, 255, 0)	Light Green	Grass
4	(0, 125, 0)	Dark Green	Trees
5	(0, 255, 255)	Light Blue	Railway
6	(0, 80, 150)	Dark Blue	Bare soil
7	(255, 150, 150)	Light Pink	Swimming Pool
8	(150, 0, 0)	Brown	Water

For **Nadam** Optimizer, following are graphs for (a) training loss, validation loss v/s No. of Epochs and (b) training accuracy, validation accuracy v/s No. of Epochs.





(a) (b)

The Classification Report is as follows:

ID	Precision	Recall	f1-Score	Support
0	0.70	0.73	.71	7206476
1	0.73	0.74	0.73	3969240
2	0.63	0.62	0.62	2758133
3	0.81	0.75	0.78	1248652
4	0.70	0.61	0.65	1542719
5	0.70	0.40	0.51	181611
6	0.90	0.77	0.83	169284
7	0.93	0.92	0.92	27473
8	0.97	0.96	0.97	1037898
Micro Average	0.72	0.72	0.72	18141486
Macro Average	0.79	0.72	0.75	18141486
Weighted Avg	0.72	0.72	0.72	18141486

Accuracy

(<u>White label</u> data is accounted)

<u>Confusion Matrix</u>:-

[[5278085	713009	640598	179626	350887	9200	7410	1910	2575]
[683417	2933473	330264	1433	1251	14065	3789	64	1484]
[723421	303792	1716861	653	2366	7478	1901	35	1626]
[266283	812	2101	936056	42758	2	640	0	0]
[562990	929	3635	32654	942216	37	0	0	258]
[22592	50322	35844	20	156	72408	32	0	237]
[17636	10806	7423	2880	390	129	130020	0	0]
[1907	271	34	6	0	0	0	25255	0]
[28101	3594	4774	0	56	217	0	0	1001156]]

<u>Kappa Coefficient (Complete Training Set)</u> = 0.6239605778685826
<u>Overall Accuracy (Complete Training Set)</u> = 71.8548083657535%

Conclusion

Satellite imagery hold a large amount of information. However, to use this information, it needs to be extracted from the raw image data. Object recognition is a method that can help the extraction of information from satellite images. Supervised object recognition approach was chosen based on the performance based on the quality of the spectral data and the number of classes. The CNN based model was then trained on the satellite images. The results based on different parameters, optimizers, learning rate and callbacks suggested that above model yields better performance with the testing dataset. This study can be explored in future research to see how the results are affected if the feature selection is done based on more number of training set and by using data augmentation.