Land Use Classification Using Satellite Imagery

Shivam Mittal, Eeshaan Sharma
2015CSB1032@iitrpr.ac.in, 2015CSB1011@iitrpr.ac.in,
Indian Institute of Technology, Ropar

Abstract

Classifying land use using satellite imagery is an important task because geo-spatial data influences critical decisions about global defense and humanitarian activities. We approached the problem using very well known convolutional neural networks, also we proposed some general techniques like resizing threshold and ensemble of CNN using the individual colour channels to decorrelate the individual learners. We obtained an accuracy of 59.24 % which is a good accuracy given the high intra class similarity between our image classes.

Introduction

Land use classification using satellite imagery is a mechanism which supports an automated analysis of satellite imagery for classifying facility, building and land use. Putting it formally, the task is to classify the bounding boxes which have been provided in satellite images into pre-defined land use categories such as - crop field, military facility, educational institution etc. The following figure shows an example image belonging to crop field class.

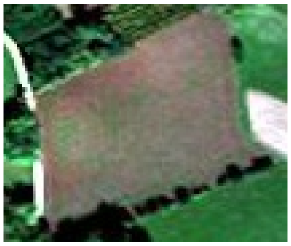


Figure 1: Satellite Image of Crop Field

The problem is motivated by the issue that intelligence analysts, policy makers, and first responders around the world rely critically on geospatial land use data to inform crucial decisions about global defense and humanitarian activities. Historically, analysts have manually identified and classified geospatial information by comparing and analyzing satellite images, but that process is time consuming and insufficient to support disaster response.

Due to rapid progress in remote sensing technologies, a lot of images of the earth via satellites are readily available. With such abundance of data, a lot of focus is being given to the problem of automatically extracting valuable information from it. The task however becomes increasingly difficult as the level of information that is to be extracted becomes complex.

The difficulty of the problem is further increased by the fact that land area images belonging to a particular class tend to show a large variability and objects may appear at different scales and orientations. This increases the within class variance of the data. Moreover there is not a significant difference between images belonging to different classes. Thus high within class variance gets coupled with low inter class distance which further toughens the problem of separating different classes and does not allow us to achieve finer classification. The following figure exemplifies the above difficulty by showing two very similar images belonging to two different classes. One of Crop Field class and other of Recreational Facility Class.

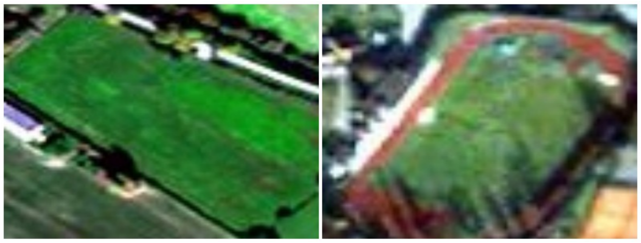


Figure 2: Crop Field vs Recreational Facility

We used a dataset containing satellite image for different land use provided by IARPA, and pre-processed it to reduce the size of the dataset as explained in the experiments section.
To perform the task of classifying satellite images, we tried a number of different approaches. First, we implemented a basic CNN, then we modified the CNN and pre-processed the data with image size threshold and then we tried the approach of using an ensemble of CNN's which trained multiple networks and then combined their outputs. Detailed description of the aforementioned approaches will be described in the following sections.

Related Works

The task of satellite image classification has been widely studied in recent times and following are some of the general methods proposed for this problem -

The first approach that we came across is a classification framework that extracts features from an input image, normalizes them and feeds the normalized feature vectors to a Deep Belief Network for classification. This approach is proposed by Basu et al[2] and is called as DeepSat or Deep Learning Neural Network for Satellite Images.

The next approach that we came across is an automated satellite image classification design using object-oriented segmentation algorithms proposed by Gamanaya et al.[3] They have deployed a region-merging segmentation technique which incorporates the spectral and textual properties of the objects to be detected and also their different size and behaviour at different stages of scale, respectively. They have linked this technique with the FAO Land Cover, Land Use classification system which has resulted in the development of an automated, standardized classification methodology.

Another technique that we came across is an Unsupervised Deep Feature Extraction technique for Remote Sensing Image Classification proposed by Romero et al. [4] They propose the use of greedy layer-wise unsupervised pre-training coupled with a highly efficient algorithm for unsupervised learning of sparse features. The algorithm is rooted on sparse representations and enforces both population and lifetime sparsity of the extracted features, simultaneously.

Methodology

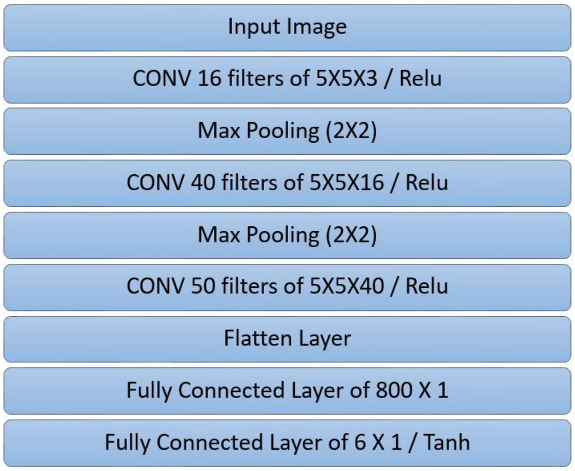
The task at our hand was tomes classify the bounding boxes present in each satellite image into some pre-defined classes. To solve the above problem we applied 3 approaches of firstly designing a basic CNN, then modifying the CNN to make use of data that we pre-processed using image resizing threshold and then training an ensemble of CNN's and combining their outputs. In this section we discuss the intuition behind why we used these approaches and why these approaches would work. In later sections we discuss in detail the experiments performed using these approaches and give a detailed analysis of the observations and results that we obtained.

Basic CNN

Since the primary challenge of the problem is to design a model powerful enough to learn the features necessary for describing the objects present in satellite images and how they are related. Hence deploying Convolution Neural Networks (CNN) is the approach that we have used to perform the task of classification. CNN's have evolved with time to best describe the images, specifically in the task of image classification and object recognition. With each layer depicting some information regarding the image, as we go deeper into the layered structure, information at each level starts holding much complex and useful relations and meaning with respect to the image. Hence the deeper layers of a CNN becomes an obvious and inevitable source to depict the characteristics of the image.

Convolutional Neural Networks are a special class of neural networks designed specifically for images. They primarily exploit the localized nature of the image wherein most of the relevant information is present in local neighboring regions. In CNN's, rather than learning weights of fully connected layers, the problem is much simplified as CNN's use convolution filters for learning fewer weights at each layer and make use of the concept of shared weights. Thus the complexity of the problem gets reduced and we are allowed to train our model on large amounts of data as that contained in the images.

The basic architecture of a CNN consists of convolutional layers where a number of filters are applied to images and their weights are learned. To introduce non-linearity in the model Relu activation function is introduced. Relu is useful due to its ease of computation. To shrink the information images are resized to a smaller size using a maxpool layer. Following this comes the fully connected layer which produces the final output. The network architecure used by us is shown in the following figure -



Input Image
CONV 16 filters of 5X5X3 / Relu
Max Pooling (2X2)
CONV 40 filters of 5X5X16 / Relu
Max Pooling (2X2)
CONV 50 filters of 5X5X40 / Relu
Flatten Layer
Fully Connected Layer of 800 X 1
Fully Connected Layer of 6 X 1 / Tanh

Figure 3: Convolutional Neural Network Architecture

CNN with Resizing Threshold

All images present in our dataset consist of bounding boxes of variable size. Before passing the images for training in the basic CNN model, the images are pre-processed and the sizes of each bounding box is resized to the average size of all bounding boxes. This might not be a good idea as we observed that there is high variance in the sizes of bounding boxes of different images.

Due to the presence of such high variance it is possible that when images of very small size are extrapolated to average size, then a lot of irrelevant information is added to them, which might lead to our model learning incorrect features. To avoid such a situation, we introduced a threshold value close to the average value of the sizes of bounding boxes. Images in which the bounding boxes had sizes greater than threshold were resized to the size of threshold and images whose bounding boxes had size smaller than threshold were resized to the size of threshold by introducing a zero padding of appropriate size. Thus the bounding box was centered in the final image and was surrounded by a padding of black (0 intensity) pixels on all sides.

The basic intuition behind using this method is that in the maxpool layer of the network, the maximum feature response is extracted from each window of appropriate size to down sample the size of the feature map obtained after performing convolution with multiple filters. Thus the zero padding introduced in images of smaller size would be automatically ignored as the image went deeper into the network because the feature response of the black pixels will be low (0 intensity).

Ensemble of CNN's

Ensemble learning is a machine learning paradigm where multiple learners are trained to solve the same problem. The generalization ability of an ensemble is usually much stronger than that of base learners. Actually, ensemble learning is appealing because it is able to boost weak learners which are slightly better than random guesses to strong learners which can make very accurate predictions.

With the possibility of achieving higher accuracy we applied the approach of training an ensemble of CNN's. Each image in our dataset is a colored image and thus it consists of 3 channels (R,G,B). We used the concept of feature bagging and trained 3 CNN's one on each channel. Thus the first CNN was trained on the Red channel for all instances, the second on green and the third on blue channel of all instances.

The intuition behind this approach is that it might be the case that for a particular class most of its distinguishing information is stored in a particular channel. Thus if that channel predicts that particular class label then it could be said with certainty that the label is correct. To accommodate information stored in all channels and to take into account all the class labels, a weighted average is taken of the outputs obtained from the 3 neural networks, as explained in the experiments section. The following figure shows the network architecture for the ensemble of CNN's used by us -

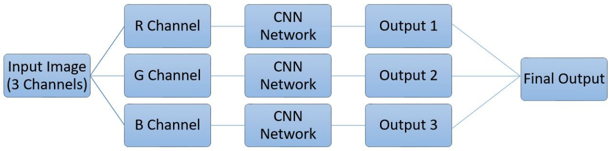


Figure 4: Ensemble of CNN's

Results and Discussion

Experimental design

Common data pre-processing The dataset used by us for training and testing our model is available under a Data Science Challenge on TopCoder in association with Intelligence Advanced Research Projects Activity (IARPA) with the name of Functional Map of the World Challenge[1]. As a part of the dataset, a point-of-interest classification library and image set is made available as training data through an AWS Public Data Set. The library includes 62 pre-defined categories and a final image set which contains 1,000,000 images each with a bounding box around an unidentified point of interest as test data. Since the size of the data is very large and due to limited computational resources, we have simplified the problem as explained bellow.
Initially the data didn't have a stratified distribution, the different classes had different number of training examples. We decided to reduce the problem to have less classes, and also to have a stratified distribution for these 6 classes so that the CNN doesn't bias towards some particular class. We considered those classes which had greater than 1500 training examples in them. When we imposed this constraint we got 6 classes which were :

- Crop Field
- Military Facility
- Educational Institution
- Place of Worship
- Recreational Facility
- Solar Farm

So, we took these classes, took randomly 1500 instances from each of these classes. Also, we took the validation data for all these classes which was provided separately. We didn't take the test data provided by them because it didn't have the bounding box information. Then, out of the images we chose, we extracted the bounding boxes out of each image for both the training and validation data. Then we split the validation data into validation and test.
From now in the report, we will consider this processed data as the basic data.
Now, we will explain the experiments for the three approaches we did.

Approach 1 - Basic CNN Since the data we have consists of bounding boxes which are themselves of varying height and width, we calculated the average height and width of all the images which came out to be 60*90. We resize all the images to 64*88 (If sizes are divisible by 2, it is easier to specify dimension after max pool layers). We did the same for the validation and training data. We saved this processed data for future training and testing.
We trained the network whose architecture explained in the Methodology section using the following techniques:

- Batch mode gradient descent (B). The impact of B is mostly computational, i.e., larger B yield faster computation (with appropriate implementations) but requires visiting more examples in order to reach the same error, since there are less updates per epoch. In theory, this hyperparameter should impact training time and not so much test performance. We chose the value 32 by empirical experimentation as makes the network converge faster as seen experimentally.
- Early stopping criteria We define a patience, i.e. the number of batches to wait before early stopping if no progress on the validation set. We chose a patience value of 50000 minibatches for training and also an patience increase of 2 if a better performing model on validation data is found, These hyperparameters are tuned experimentally in a such a way, that they ensure that neither the network trains too long when we know no more learning is taking place (the validation accuracy being is achieved is less than best achieved in the training so far). But the patience is large enough to make the network be able to escape any local minima.
- Momentum We are using a initial momentum of 0.1 and increase it by 5after every 10 epochs. This is done to make the updates proportional to the smoothed gradient estimator instead of the instantaneous gradient g. The idea is that it removes some of the noise and oscillations that gradient descent has, in particular in the directions of high curvature of the loss function
- Adaptive learning rate The parameters are chosen by experimental testing to give best accuracy on the validation data, so we are basically scheduling our learning rate to fit our dataset's characteristic best.

We saved the model which was giving minimum error on the validation dataset, and used it for testing. We will explain the results of the experiment performed in the later section.

Approach 2 - CNN with resizing threshold In this we had used a image resizing threshold because of the intuition described in the methodology section. This approach has training and testing procedure almost similar to the first approach (all the techniques batch mode, early stopping, momentum and adaptive learning rate were used and tuned accordingly), just the data processing step is changed in which we considered the resizing threshold to be 64*64, this was considered as it was close to the average size of the bounding boxes size, and also is a perfect multiple of 2, so defining max pool layers becomes convenient. Now, for any image in training, validation or test, if the image size is greater than the threshold (64*64), we will resize the image down to 64*64 and save it. But, if the image has size smaller than 64*64, we do not upscale it, rather we just place the smaller image at the center of a 64*64 image which has rest of pixel intensities as 0. The intuition behind doing this is explained in the methodology section above.

Approach 3 - Ensemble of CNN We processed all the image in the training, validation and test data by extracting the different channels (R, G and B) of the images and saving them separately as data for different networks. Then, we trained 3 different models. Model 1 was trained on the red channel of all the images, model 2 was trained on the green channel of all the images and model 3 was trained on the blue channel of all the images.

For testing we extracted the channels individually and fed to the corresponding model which was trained on that channel, and obtained 3 different labels. The accuracy was measured separately for each network. Then for the final output of the ensemble, we took the majority vote initially, but it gave less accuracy than the red channel. So, we computed the confusion matrix for each of the networks, and then combined the final result by taking the class label from that model which was the most accurate in predicting the class. This confusion matrix calculation and combining the result was done purely on the basis of the validation data, so the result on the test data is unbiased. We will explain the exact method for combining the result in the results section because there we will also present the confusion matrix to give a better idea.

Results and Observations

The accuracies of the approaches we tried have been given in the table below :

Table 1: Test accuracies of the different approaches

Approach	Test accuracy
Approach 1 - Basic CNN	59.242424
Approach 2 - CNN with resize threshold	49.21875
Approach 3 - Ensemble of CNN	55.90

We observe that that the test accuracy of the basic CNN is the highest. This is understandable because CNN are very efficient for learning feature necessary for the task, and they give an accuracy of 60% which can be considered as a good accuracy because there is a lot of similarity between the classes we have chosen, and the task is a difficult classification task.

The CNN with resizing threshold didn't give good results which might be because of the case that the small size images which are being placed at the center too are of different sizes. So, in one image there might be majority black pixels and our filter weights get biased to reduce the filter response from those pixels, and less attention is given to the actual image being placed at the center which contains the relevant information. So, this might cause a feature representation to be learnt which is not dicriminative enough.

The ensemble of CNNs gave good result but not exceptionally good result. We will discuss the individual accuracies of the individual trained CNNs on the different channels, and show their confusion matrices to see where each CNN is making a mistake, and our approach to construct the final ensemble.

We can see that overall the accuracy of class 1 and 2 is not very good. Class 1 is being confused with class 5, i.e, crop field is being confused with recreational facility which are very similar looking, as also shown in the introduction of the report. Also, class 2 is being confused with class 3, i.e, military facility is being confused with educational institute because the images for both of the classes look like

Table 2: Confusion matrix on red channel CNN

Class	1	2	3	4	5	6
Class 1	46	14	3	5	30	12.
Class 2	10	49	35	10	1	5.
Class 3	6	13	67	17	4	3.
Class 4	2	1	12	86	8	1.
Class 5	23	3	4	13	66	1.
Class 6	14	9	12	7	8	60.

Table 3: Confusion matrix on green channel CNN

Class	1	2	3	4	5	6
Class 1	58	20	1	2	20	9
Class 2	9	61	25	10	3	2
Class 3	10	17	60	18	3	2
Class 4	4	7	6	87	6	0
Class 5	32	7	3	15	52	1
Class 6	22	17	9	6	5	51

Table 4: Confusion matrix on blue channel CNN

Class	1	2	3	4	5	6
Class	42	10	10	7	24	17
Class	8	52	29	9	1	11
Class	3	12	69	18	3	5
Class	0	3	10	82	14	1
Class	12	6	6	18	64	4
Class	21	11	10	5	10	53

Table 5: Accuracy of the different channel CNNs

CNN	Test accuracy
Red channel CNN	55.60
Green channel CNN	51.96
Blue channel CNN	50.90
Ensemble of the three	55.90

a collection of individual building and they are not easy to differentiate.

We see that the accuracy of the red channel is the highest, so we can conclude that overall the red channel has the most discriminative information out of all the colour channels. We see that the blue channel has the lowest accuracy and we can say that it does not have that much discriminative information in it. Although we can also see that the green channel predicts class 4 with very high accuracy. So, using these observations which we obtained from the validation data, we constructed a function for obtaining the final output of the ensemble from the predicted labels of individual networks. We take label 4 if it is being predicted from network 2 (high accuracy on class 4), otherwise we take the label prediction from network 1 (high accuracy on other classes). Finally using this ensemble function we can see that the accuracy on the test data is increased for the ensemble classifier, although marginally. We can also say that since we de-correlate all of our networks by considering only 1 channel, we reduce the data at the same time, and hence this might be a reason of lesser accuracy for the ensemble.

Summary

In short, we can summarize that we approached the problem of land use classification using satellite images using well known convolutional neural networks. We proposed some general techniques like resizing threshold and ensemble of CNN using the individual colour channels to decorrelate the learners. These techniques provided average results in comparison to the basic CNN. But, we cannot generalize that these techniques will not work for land-use classification because we were working with limited data which is not enough for generalization. Overall, we got decent accuracy.

Future Work

If we had more time and computational power, we could have used more data, and also more deeper networks which are able to learn more complex features necessary for the classification task. We also limited the input size of the image we feed to the network because we had memory errors on the machine we were running our experiments on. Larger image size means less loss of information which might give better accuracies.

Acknowledgement

We would like to express a note of thanks to our professor Dr. Narayanan C Krishnan and our Teaching Assistants Mr. Sanatan Sukhija and Ms. Akanksha Paul. Without their constant guidance and support this project would not have been possible.

References

[1] Functional map of the world challenge - (www.iarpa.gov/challenges/fmow.html).

[2] S. Basu, S. Ganguly, S. Mukhopadhyay, R. DiBiano, M. Karki, and R. Nemani. Deepsat: a learning framework for satellite imagery. In *Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems*, page 37. ACM, 2015.

[3] R. Gamanya, P. De Maeyer, and M. De Dapper. An automated satellite image classification design using object-oriented segmentation algorithms: A move towards standardization. *Expert Systems with Applications*, 32(2):616–624, 2007.

[4] A. Romero, C. Gatta, and G. Camps-Valls. Unsupervised deep feature extraction for remote sensing image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 54(3):1349–1362, 2016.