INTER IIT TECH MEET - (2018-19)

EYE IN THE SKY

INDIAN INSTITUTE OF TECHNOLOGY, GANDHINAGAR

Team Members:
Mrinal Anand - mrinal.anand@iitgn.ac.in
Contact no - +91-8511422558
Rushil Shah - shah.rushil@iitgn.ac.in
Contact no- +91-9426700450
Rithwik Kukunuri - kukunuri.sai@iitgn.ac.in
Contact no- + 91-7433064232

Abstract -

This Report describes our approach to the Eye In The Sky challenge as a part of Inter IIT Tech Meet (2018-19). The primary objective of this challenge is to accurately segment the satellite images as grass, roads, buildings, soil, trees, railways, water bodies, swimming pools.

Introduction -

Satellite images are rich source of information and plays a vital role in providing geographical informations. Satellite and remote sensing images provides quantitative informations that reduces complexity of the field work and study time. The volumes of data receive at datacenters is huge and it is growing exponentially as the technology is growing at rapid speed as timely and data volumes have been growing at an exponential rate [1]. There is a strong need of effective and efficient mechanisms to extract and interpret valuable information from massive satellite images. Satellite image classification is a powerful technique to extract information from huge number of satellite images. Satellite image classification is a process of grouping pixels into meaningful classes and classify each part at the pixel level into one of the predefined classes. This classification technique is used for Spatial data mining, Thematic map creation, visual and satellite image interpretation, Field survey and Disaster management [2]. Besides the practical need for accurate aerial image interpretation systems, this domain also offers scientific challenges to the computer vision. We explore the challenges faced due to the small size of the dataset, the specific character of data, and supervised and unsupervised machine learning algorithms that are suitable for this kind of problems.

Classification approach -

Motivation - Before the CNN came into the picture, Image processing techniques were widely used for segmentation of satellite images. When we take an Image processing technique based approach for segmentation of satellite images, we need to come up with an algorithm that does justification for all the kinds of images possible. In this era, where data is vastly available, CNN based approach will provide a good generalized algorithm for the segmentation task. The current state of the art in computer vision is CNN (Convolution neural network) which is a supervised technique. And the successful state of the art technique in semantic segmentation is variants of FCN (Fully convolutional network). The essence of this approach is to use a CNN as a powerful feature extractor while replacing the fully connected layers with convolution one to output spatial maps instead of class scores. Those maps are upsampled to produce dense per pixel output [5].

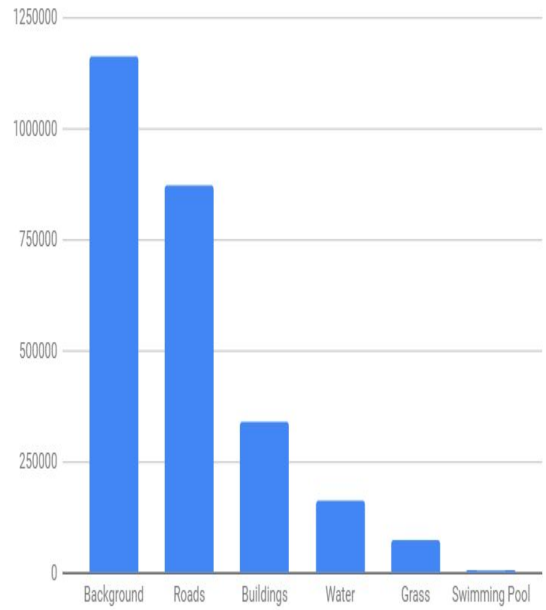
Methodology -

Our basic solution to this challenge was based on modified fully convolutional neural network architecture called U-Net. This architecture was made to tackle the problem of image segmentation in biomedical images. are suitable for this kind of problems.The U-Net architecture allows combining low-level feature maps with higher-level ones, which enables precise localization. A large number of feature channels in upsampling part allows propagating context information to higher resolution layers. This type of network architecture was specially designed to solve image segmentation problems effectively [3].

Dataset -

The data set that was provided consist of 14 satellite images and its ground truth values.The images were is ".tif" format which is able to store high resolution data without loss. The images are of different sizes and total of 8 class were there namely- Building, Roads, Tree, Grass, Bare soil, Water, Railways and Swimming pool. The given image has 4 channels (RGB +NIR). The images were highly imbalanced while taking the class distribution into account.

Visualising satellite images using QGIS



Our initial approach was to create a model for differentiating a Class K vs everything else (Binary classification for each class). This way we will end up with 9 Models(including the background class). But since our dataset was largely imbalanced for most of the classes (1:10) resulting in high accuracies, but low precision.

- To improve the model we create a custom loss function which heavily penalize the model if there exist a misclassification of positive class with a negative class.
- We tried jaccard index (also known as intersection over union) as primary evaluation metrics to improve the accuracy. But it didn't work out well.

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}$$

$$0 \leq J(A,B) \leq 1$$

Unsupervised Method : Reflectance Indices -

The fact that we have infrared and other channels from the non-visible frequency range allows us to identify some class purely from the pixel values, without knowing any contextual information. Where CCCI index and NDWI index is used for the detection of water. We differentiated swimming pool and waterways by setting different threshold values (0.7511 for swimming pool and 0.7511-.576 for water). We used Hyperopt [4] for optimizing the parameters, after that we used morphological transformation (erosion and dilation respectively) to reduce the number of false positive labels. The reason behind using morphological transformation is to remove the false positive values that are coming from the building areas.

$$\mathrm{CCCI} = \frac{\mathrm{NIR} - \mathrm{RED}_{\mathrm{edge}}}{\mathrm{NIR} + \mathrm{RED}_{\mathrm{edge}}} \times \frac{\mathrm{NIR} + \mathrm{RED}}{\mathrm{NIR} - \mathrm{RED}}$$

$$\mathrm{NDWI} = \frac{\mathrm{GREEN} - \mathrm{NIR}}{\mathrm{GREEN} + \mathrm{NIR}}$$

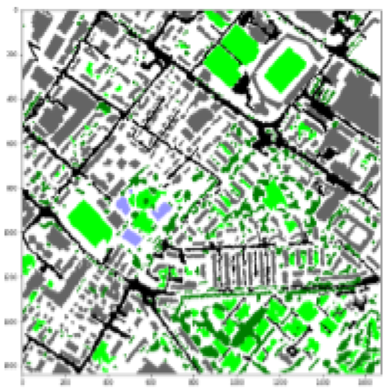


Fig 1-- Ground truth mask

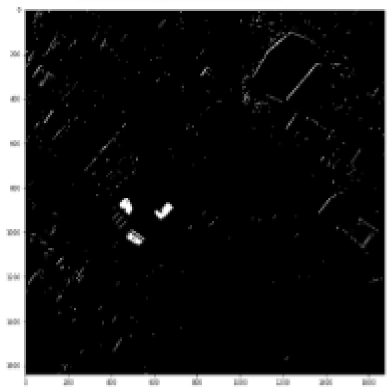


Fig 2 Result after Thresholding

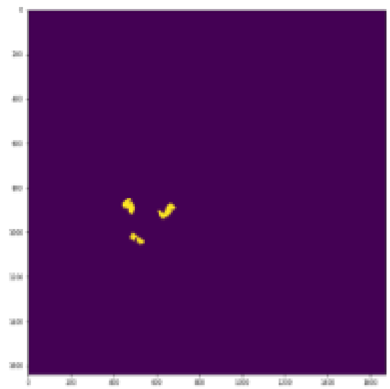


Fig 3 Mask after erosion and dilation

Multi spectral Unet-

While doing online research, we found out that U-Net and its variant are being frequently used for image segmentation. So initially, we were experimenting with the basic U-Net architecture for our task. All the variants of the above model we having an approximate accuracy of ~78%. On plotting the confusion matrix, we observed that buildings were misclassified as roads and vice versa. We also explored the other variants of U-Net by adding Resnet Blocks, creative activation functions, optimizers, custom loss functions, and the results weren't amazing. Then we came across an idea of feeding in the edges detected to the neural network, hoping that it can find a difference between roads and buildings. The model started to show some improvement. Still the results weren't satisfying. By default U-net is very deep. After passing through many layers, the key information present in the raw image is lost, especially the edge detection features. To overcome this issue, we fed in the edge detection channel directly into the last layer after feeding it through a Convolution filter. We did the same thing for NIR channels, as they have few raw features that can be used for the task of segmentation.

We trained around 30 models with different optimizers, initializers, activation function, evaluation metrics to find the optimal parameters for our model.

- Activation function - After using several activation function we founded out that ELU is more resistant towards noise than other activation like Relu, PRelu, LeakyRelu.
- Initializers - We initialized the values with ‘he_normal’.
- Evaluation metrics - We monitored the accuracy and jaccard index.
- Filters - To generate new feature we applied Sobel Filter, Kenny edge detection, Robert Filter, morphological transform and local binary transform (to detect texture in the image).
- We tried Dilated convolution layer to increase the receptive fields but its accuracy was saturating

Before feeding in the images to neural networks we tried different normalization function to make data more gaussian. We found out normalizing the data between 0 to 1 gives better results than normalizing between -1 to 1. We initially used power transform function from sklearn library to normalize the data between 0 to 1 but it was computational very heavy so we did an approximation but taking log of it and then dividing it by log of maximum of array.
Due to hardware limitation we have restrict the size of image that we are feeding into the network, so we divided the images into small patches. We prefered to maximise batch size over the patch size because it significantly increases the receptive field.

We also tried out mask rcnn architecture which is state of the art in image instance segmentation. But due to limit amount of data the model was overfitting.

Mask of edge detection after applying sobel edge filter

Final Network is as Follows:

- Modified U-Net with Exponential Linear Unit activations and Relu Activations with Batch Normalization Layers.
- A Parallel layer that processes the NIR channel and is concatenated before computing the Softmax.
- Another Parallel layer that processes the detected edges(sobel edge) and is concatenated before computing the Softmax.

Pre-Processing:

- Each image is rescaled between 0 and 1.
- Then each image is made into 96x96 patches.
- Each patch has 5 Channels in it. Last channel is computed by doing the sobel edge detection filter.

Hyper Parameters:

- Trained for a total of 100 Epochs.
- First forty epochs were trained using Adam Optimizer. Next ten epochs were trained with SGD with learning rate of 0.0001.
- Batch size was 32.

Implementation -

- We used Keras framework backend of tensorflow to implement our model.
- We are doing k fold One Out Cross Validation. In total we will be training 14 different models and these models will be optimized by using 13 images and its performance is evaluated on the 14^{th} image.
- Edge detection is done using filters in skimage library.
- Reading and writing the image in tiff format is done using tifffile library.

Results:

Metric	**Value (Mean)**
Accuracy	.93
Kappa Coefficient	.9028

Total Confusion Matrix:

The given confusion given below **does not include the prediction of background class.** We have considered only 8 classes. Also the given confusion matrix includes prediction of all the 14 images (after applying k fold validation, where we have taken k=14).

	Road	Bare Soil	Green Trees	Green Grass	Buildings	water	Swimming Pool	Railways
Road	2492251	19370	32107	12642	169020	4781	11	27776
Bare Soil	6256	146032	1293	9716	2862	734	0	2374
Green Trees	18430	41	1477457	36996	9066	279	0	416
Green Grass	7326	5418	46579	1184478	4700	79	1	59
Buildings	240728	6406	14410	10277	3692092	1100	52	3915
water	17156	14	1582	126	7084	1008057	0	3840
swimming pool	968	0	110	156	2255	345	23593	0
railways	67187	26	1822	295	52289	455	0	59506

Columns: Prediction
Rows: Truth

Please find the Computing_Metrics.ipynb notebook for image wise confusion matrix.

References -

[1] Zheng, X., Sun, X., Fu, K. and Hongqi Wang, 2013. “Automatic Annotation of Satellite Images via Multifeature Joint Sparse Coding With Spatial Relation Constraint”, IEEE Geoscience and Remote Sensing Letters, VOL. 10, NO. 4, JULY 2013, pp.652-656.

[2] Sunitha Abburu, Suresh Babu gola, 2015. “Satellite image classification methods and technique: A review” ,International Journal of Computer Applications 2015.

[3] Ronneberger, Olaf, Philipp Fischer, and Thomas Brox. ”U-net: Convolutional networks for biomedical image segmentation.” International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer International Publishing, 2015.

[4] https://github.com/hyperopt/hyperopt

[5] Vladimir Iglovikov, Sergey Mushinskiy, Vladimir Osin, 2017 “Satellite Imagery Feature Detection using convolutional Neural network”, Arxiv, 2017.