Detection of Red Cars from High Resolution Satellite Imagery

Nikhil Reddy Kortha, *University of Florida, Gainesville, Florida, nikhil.kortha@ufl.edu*

***Abstract* — Identification of specific objects from high definition satellite images is a very interesting and challenging problem with applications spanning across several fields like remote sensing, military, traffic monitoring, etc. This project aims to identify red cars from high resolution satellite images. RGB values for every pixel are used as featured to train the KNN classifier. A preprocessing step of transforming the huge training data to the fit the model is presented in this paper. The training image is split into several sub-images based on the ground truth due to heavy volume of pixels in training image. The trained model is used for testing and result is the location of red car in the test image.**

***Index Terms*— Object detection, KNN Classifier, supervised learning, High resolution satellite imagery**

I. INTRODUCTION

Nowadays, the task of detecting objects from satellite images has become very important because of its applications in a wide spectrum of domains. They can be used by the military to guard the borders, have surveillance on enemies [1]. They can also be used for remote sensing applications, traffic monitoring, disaster mitigation and planning etc. [5] Detection of small objects of interest from a huge image still remains a challenging task. In this project, I propose a method to detect red cars from a high definition satellite image. First, the image is preprocessed and split into several sub images of the training data based on regions of interest. Then, the RGB values of every pixel will be used as features to model the KNN classifier. As we have the initial training data with labels to detect red cars, this problem is now a supervised learning task, where we classify each pixel to be either red or not red. This kind of approach of object detection using colors was used in the field of robotics by Hani Hun ud [2] where he used color-based detection to detect ground object sin emergency situation by Unmanned Air Vehicles. Pedro M. R's [3] paper on survey of color spaces and color-based segmentations to detect color coded objects gives us some insights on how the RGB band along with segmentation can be used to detect very small imagery.

The main challenge here is the large swaths of area in the image, which makes the training very difficult. This is because, we are detecting a single object of a specific color and due to the large background spaces and huge number of pixels, these specific target objects will be in minority. The data will not be enough to detect the red cars. The preprocessing step, which divides this huge training image into small sub images around the target object (red cars) reduces the number of pixels to be trained thereby reducing the computational time.

This paper is defined in the following manner: Section II contains the implementation details of the proposed project. Section III gives a detailed explanation of how this experiment is performed with results. Section IV concludes this project by giving scope to the future work that can be formed on this experiment.

II. IMPLEMENTATION

The main idea behind this method is to preprocess the training image and identify regions of interest from the training data i.e. places where the red car is present and train the model with these data. sub images of size m*m were constructed around the red cars and trained the model with these images. This is because, Firstly, the time to train the complete image takes a lot of time and secondly, the red cars which we are interested to train our model are very few.

In this project we are provided with ground truth, the location of red cars in the training image and we need to find the red cars in testing image. Since, the training image is of size 6250 X 6250 and sending all 38 million pixels for training would be a complex task, a preprocessing step is performed on the training data.

The implementation of this approach is shown in **Figure 1**. In the first stage the training image is split into sub images of size m X m where red cars are present. This is done with the provided ground truth. If the coordinates for ground truth are x, y then the image is split in such a way that the sub image m x m has x, y as midpoint. The sub image is formed from the following coordinates as its boundaries: x-m/2, x + m/2, y-m/2 y + m/2. The total number of sub images built from this approach are equal to the number of values in ground truth. In the second stage the sub images will be used for the training and validation purpose. The RGB values of these pixels are used as features for the training of the classifier. We split the data such that 70% of these images are used for training the KNN classifier and the remaining 30% of the images are used for validation. After validation for various m values, we decide the approximate m, k vales and we re-train the model to find the location of red cars in the test image. The experimentation

results and accuracy are discussed in the section III.

A. *Cross Validation with p folds*

Cross validation is a popular technique used to resample the training data. A single parameter p determines number of groups the data should be spilt into. This technique helps us estimate the robustness of the model in predicting the unseen data. The value of p determines the bias in the data.

For this experiment we choose p as 10, as these values, empirically yield test error rate which balances both bias and variance.

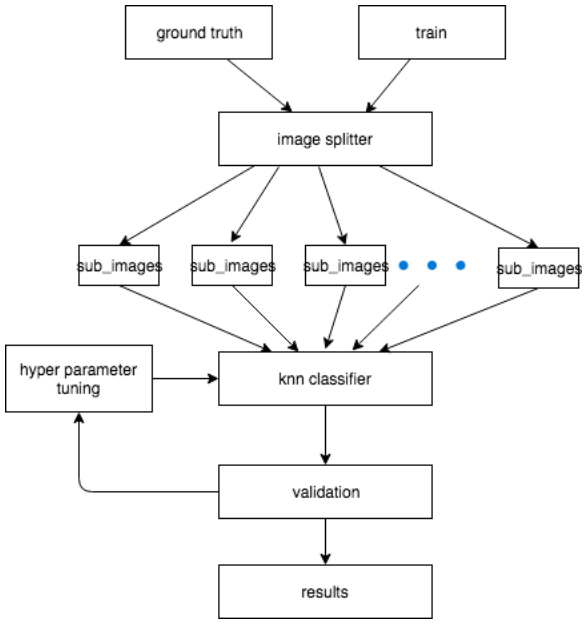

Figure 1

III. EXPERIMENTATION AND RESULTS

As discussed in the above implementation, in the preprocessing step, the training data is constructed by building sub images with the ground truth as midpoint. The training image is split into smaller images because it takes a lot of computational time to train the model on the whole image. A sample of sub images is shown in Figure 2. 70% of these smaller images were used for training the model and the remaining 30% of them were used for validation. A 10-fold cross validation is performed on the 70% of the training data to improve the robustness of the model. After performing the cross validation, we do hyper parameter tuning and select the ideal number of neighbours required in the K nearest neighbours algorithm. Here, the accuracy depends on both k value and m value.

From Figure 3, we can see that, lower the value m, lower the accuracy. This is because, if the m value is too low, there are not enough pixels present for the classifier to learn and classify. For m > 50, the accuracy remains the same. This is because, for m > 50, we will have enough pixels to train the classifier and classify the pixels without any confusion. After performing various experiments, for m value between 50 – 100, the results turn out to be accurate.

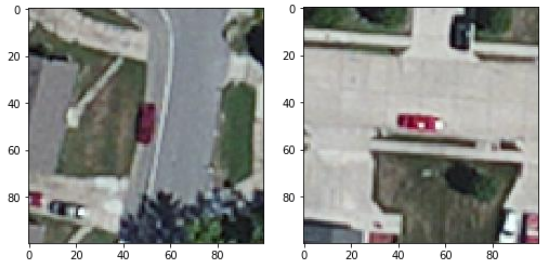

Figure 2

Accuracies for different k values:

M	K	accuracy
100	1	0.99
100	2	0.99
100	3	0.99
100	4	0.99

We observe from the above table that, no matter the k value, the accuracy is 0.99 this is because the classifier is classifying everything as not a red car and still the accuracy is very good as the number of cars are very less. So as the standard accuracy metric does not give a good picture of the correctness of our model we define a new accuracy metric as custom_accuracy as the ratio of predicted red cars to the actual number of red cars.

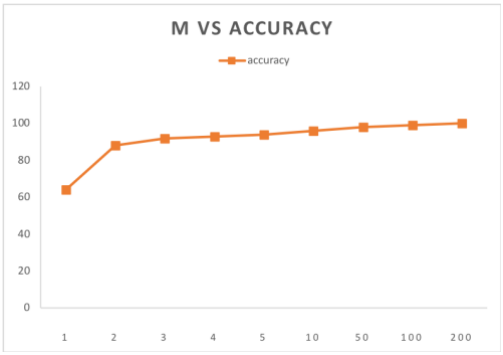

Figure 3

For, m in the range of 50 -100, we performed various experiments to determine the ideal value of k to be used in the K nearest neighbors algorithm. We found that, for k between 3 to 4, value of custom_accuracy to 1. This means that, for k between 3 – 4, the number of predicted red cars are close to the number of actual red cars in the training data. The test mage is shown in Figure 4. After performing the above algorithm on the test image, the results are shown Figure 5. From Figure 5 and Figure 6, we can see that the red cars have been detected.

Experimental results are shown in the below tables:

For 200*200 sub image

K	Predicted # of cars	Actual number of cars
2	3	7
3	4	7

4	5	9

For 40*40 sub image

K	Predicted # of cars	Actual number of cars
2	1	12
3	2	12
4	2	7

The test file contains all the coordinates of pixels of the red car. Post processing can be applied to combine all the detected pixels.

A. Why K nearest Neighbors

K nearest neighbors is a simple approach. As it is a non-parametric approach, we do not pick up any distribution. The given data is hugely distributed, and the cars are sparse. So, there is a lot of variance, and hence it becomes difficult for other algorithms to detect red cars. In probabilistic generative classifier, we don't actually work on real data, we calculate mean and variance. For this kind of data, both the mean and variance would be biased. For linear regression, we have weight values as parameters and calculating weight values for more than 1000 pixels would increase the training time. KNN algorithms has many disadvantages too. For one test instance, we have to calculate the distance metric from every training sample. If the sample is huge, and the test sample is very small, this method is not useful. Considering the high number of pixels and distribution of data, I chose KNN algorithm for implementing my method.

Figure 4

Figure 5

Figure 6

IV. CONCLUSION

Training the model for huge amounts of data is very complex. So, in this project, I presented a transformation step of training data, which reduces the volume of the pixels without loss of any important features. Though this model predicts the location of red cars from the image it considers other objects which are red as a red cars. Future work can be done to solve this problem and also this project by using KNN with a different distance metric or with spatial constraints can be used. The performance of this project can be enhanced if neural networks, convolution neural networks [4] are used.

V. REFERENCES

[1] I. Bayir, "A glimpse to future commercial spy satellite systems," *2009 4th International Conference on Recent Advances in Space Technologies*, Istanbul, 2009, pp. 370-375. doi: 10.1109/RAST.2009.5158227

[2] Kadouf, Hani Hunud A., and Yasir Mohd Mustafah. "Colour-based object detection and tracking for autonomous quadrotor UAV." *IOP Conference Series: Materials Science and Engineering*. Vol. 53. No. 1. IOP Publishing, 2013.

[3] Caleiro, Pedro MR, António JR Neves, and Armando J. Pinho. "Color-spaces and color segmentation for real-time object recognition in robotic applications." *Electrónica e Telecomunicaçoes* 4.8 (2007): 940-945.

[4] X. Chen, S. Xiang, C. Liu and C. Pan, "Vehicle Detection in SatelliteImages by Hybrid Deep Convolutional Neural Networks," in *IEEE Geoscience and Remote Sensing Letters*, vol. 11, no. 10, pp. 1797-1801, Oct. 2014.

[5] G. Cheng, P. Zhou and J. Han, "Learning Rotation-Invariant Convolutional Neural Networks for Object Detection in VHR Optical Remote Sensing Images," in IEEE Transactions on Geoscience and Remote Sensing, vol. 54, no. 12, pp. 7405-7415, Dec. 2016.