K-Nearest Neighbors Approach for Classifying Red Vehicles in High Resolution Satellite Images

Diandra Prioleau

***Abstract*—The use of high resolution satellite images with vehicle detection methods is vital in many areas in the world, most specifically in traffic monitoring. Therefore, it is important to continue to improve vehicle detection algorithms to increase the detection rate and to lower execution time and the risk of false alarms. Using high resolution satellite images, this paper discusses the implementation and experiments conducted, in which k-means was employed to determine variations of the RGB values in the background and in the red pixels of the vehicles in the ground truth data to create classes representing the variations for classifying red vehicles using the k-nearest neighbors algorithm. Using a 70/30 split on both the background data and ground truth, which contains the pixels of the red vehicles, a validation was done on the training data, which achieved a 96% classification accuracy and 65% classification accuracy when validating only with the 30% from the ground truth data.**

I. Introduction

VEHICLE detection is a useful tool in today's society, where it is used for traffic monitoring and surveillance for law enforcement and military purposes. Significant research has been done in this area to develop algorithms that have high performance with low execution time. In this section, a literature review of previous works with vehicle detection and color recognition will be discussed. Afterwards, an overview of the experiment for this project will be given, along with a brief discussion of which implementations from the literature review were considered in the implementation for this project.

A. Literature Review

Wei et.al developed an automatic vehicle detection method using edge detection and a morphological method, which erodes and dilates the image, to identify vehicle pixels in aerial images from a parking lot [2]. In addition, they use a self-adaptive algorithm to detect vehicles by creating a template of a vehicle with 12 models representing different orientations of the vehicle. The correlation coefficient, which determines whether the object in the image is like the template, must be greater than or equal to the threshold in order to classify the object as a vehicle. From their experimentation, they discovered that their system only needed a small number of training samples to produce sufficient results in comparison to other object detection systems [2].

Zheng et. al used k-means and mean shift for color recognition of clothes in images [3]. They utilized k-means to extract the background and foreground of the image, which is an image of an individual wearing clothes of some color with a complex background. Then, they used mean shift to cluster the three-dimensional vector representing the RGB values and "to count the maximum clustering class pixels and give the color name of that image by getting the minimum distance of calculating it with" a standard clothing color database [3]. Combining k-means and mean shift resulted in a performance above 85% for color recognition. However, if the background and color of the clothing is similar, it is becomes difficult for k-means to separate the background from the foreground.

Cheng et.al developed an automatic vehicle detection system for aerial surveillance using pixelwise classification [1]. They considered vehicle color features and local features (corner and edge detection). Vehicle colors were used to distinguish between vehicle colors and non-vehicle colors. The local features involved detecting edges and corners in the image to possibly identify vehicles. After which, the pixelwise classification was performed using a dynamic Bayesian network (DBN) to classify a pixel as a vehicle or non-vehicle pixel and compared with a Bayesian network (BN). It was found that removing the background and using the edge detector resulted in a higher hit rate, or increase in vehicle detection, and lower false alarms than when either one or both methods are not implemented. When comparing the processing speeds of the framework with other methods (MVDRD, Cascade, Symmetric), both BN and DBN outperformed Symmetric but required more processing time than MVDRD and Cascade. However, if high detection rate and low false rate are the primary considerations , then the execution time achieved by BN and DBN, which was approximately 1 second per frame, is still sufficient in performance.

B. Overview

The goal of this research is to develop to a machine learning approach to detect red vehicles in a high resolution satellite image. After doing a thorough literature review, it was determined to use k-means and k-nearest neighbors (KNN). As Wei et.al did not report the results of the other object detection systems [2], this methodology was not considered for implementation in my experiment. The approach used by [3] to separate the background and foreground using k-means was implemented in my experiment, but instead of using k-means to separate the background and foreground, it was used to determine the variations in RGB values for the background and the red vehicles given in the ground truth data. After determining the variations of the background and ground truth data, a pixelwise classification was conducted, similar to [1], but instead of using a Bayesian network for predicting whether or not a pixel belongs to the red vehicle class, the k-nearest neighbors algorithm was utilized.

II. Implementation

Here, we elaborate in detail the different steps taken in classifying red vehicles in high resolution satellite images. This implementation was chosen based on the pros and cons of the algorithms permitted to be used. Since time consumption was a high priority, using the K-Means was chosen for pre-processing the data. In addition, since the images are large in data, the k-nearest neighbor (KNN) algorithm was a good option for predictions since it is most effective when the training data is large. However, since the disadvantage of KNN is that it has a high computation cost, the image data needed to be reduced to minimize time. Therefore, it was necessary to be mindful of the size of training, validation and test data sets that would be passed to the KNN classifier. Using linear regression wis not the optimal choice for classification problems, such as this one, and the probabilistic generative (PG) classifier was not the optimal choice either in comparison to KNN because KNN, which is a non-parameteric method, does not assume a generating distribution, which if chosen poorly, may result in a poor model of the distribution; therefore, the PG classifier was not chosen.

A. K-Means and KNN

K-nearest neighbors is a non-parametric, machine learning algorithm that compares a test point to each data point in a training set using some specified distance metric. There are several distance metrics, such as the Euclidean distance, city block distance, Mahalanobis distance, etc., that can be used with KNN. The Euclidean distance was used in the experiments since color distance calculates the Euclidean distance in color space using the RGB values, where

$$d_E = \sqrt{(R_a - R_b)^2 + (G_a - G_b)^2 + (B_a - B_b)^2} \qquad (1)$$

It then determines the "k" number of training data points that the test point is closest to based on the distance. The test point is then given the label of the class with the most data points represented in the set.

K-Means is a clustering algorithm, in which a specified number of clusters are given . The objective function, given below, "represents the sum of the squares of the distances of each data point to its assigned vector", where the goal is to find the $r_{n,k}$, where $r_{n,k} \in 0, 1$, and μ_k, which is associated with k^{th} cluster, that minimizes J [4].

$$J = \sum \sum r_{nk} \|(x_n - \mu_k)^2\| \qquad (2)$$

B. Pre-processing Data

1) Training Data: The ground truth, which was provided by a student, Bo Hu, from the Foundations of Machine Learning course, was first processed to extract the RGB values from the coordinates given for the pixels of the red vehicles in the image. Likewise, the RGB values of the background data in the image were extracted by determining the dark spots in the image. It is possible that some of the data extracted for the background could include vehicle and possibly red vehicles. However, the dark spots were plotted on the image, shown

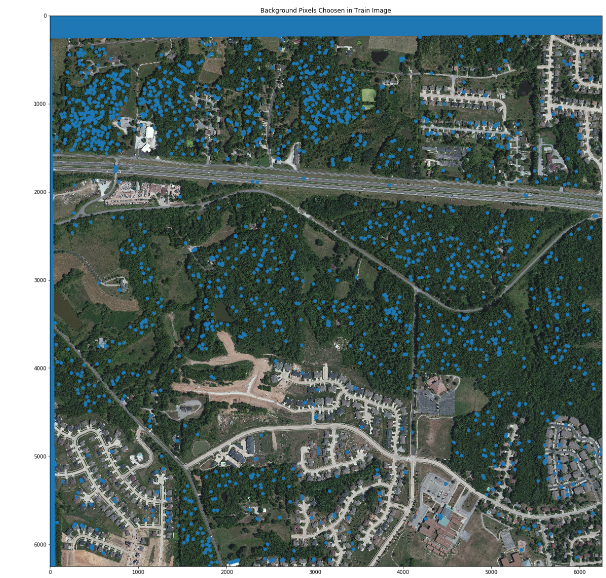

Fig. 1. Coordinates of Background in Train Image

in 1, and examined to ensure that little to none of the pixels chosen were those of vehicles in the image.

Afterwards, similar to [3], the k-means algorithm was implemented separately for the red pixels and background pixels to determine the variations in RGB values. For example, there are multiple shades of red. Using the k-means algorithm allowed for us to learn 3 separate cluster centers that represent the red vehicle class and similarly to learn 5 separate cluster centers for the background class to account for variances in the classes. In addition, K-Means was used to label the ground truth of red vehicles and one-fourth of the background in the imagery. Only one-fourth of the background pixels were labelled to decrease time consumption. These background pixels were combined with the ground truth data to create the training data.

2) Test Data: The approach discussed in [1] was considered to pre-process the test image. As part of their vehicle detection framework, Cheng et.al removed the background. However, instead of removing the background, the reverse was done by extracting the pixels that were red from the image. Next, for feature extraction, the corners in the extracted red pixels were detected. It was assumed that if a pixel had a RGB value close to red and was a corner, then it may be a red vehicle. The red pixels that were also chosen as corners were used as the test data that was predicted with the KNN classifier.

C. Training and Validation

A 4-fold cross validation was conducted using the ground truth data and background data for k values of 1 - 25, only the odd numbers, to determine the optimal k value for the k-nearest neighbors algorithm (KNN). A 4-fold cross validation was conducted on both the ground truth data and background data. After which, the first fold from the ground truth data and that from the background data were combined together, respectively, to form the training-validation data set. This was done for all 4 folds. The results of the cross validation is shown in Table I. From this, it was determined that the optimal k was 19 because it achieved the highest classification accuracy and its variance of the classification accuracy for the 4-fold cross validation was low, essentially being zero.

TABLE I
4-FOLD CROSS VALIDATION RESULTS

k	Mean of Accuracy	Variance of Mean Accuracy
1	0.959162	4.765917e-07
3	0.966408	3.863537e-07
5	0.967849	1.307300e-07
7	0.968584	7.540184e-08
9	0.968865	1.424168e-07
11	0.968949	2.637444e-07
13	0.969107	4.716424e-07
15	0.969181	3.393939e-07
17	0.969171	4.483040e-07
19	0.969311	2.203100e-07
21	0.969248	3.202593e-07
23	0.969133	4.456066e-07
25	0.969145	1.080538e-07

D. KNN Approach

Cheng et. al conducted a pixelwise classification using a Bayesian network and a dynamic Bayesian network[1]. Similarly, a pixelwise classification was conducted but instead of using a Bayesian network, the k-nearest neighbors algorithm (KNN) was employed. KNN was chosen as the algorithm for predicting which class each pixel in the validation and test data sets belonged because it is easy to implement but mostly because the number of red vehicles in the image is small in comparison to the background; therefore, since k-means was being used to determine variations of RGB values for red pixels and the background, using KNN would be more effective than other algorithms in determining how close the pixels in the validation and test sets are to the those in the training data because the KNN classifier has been fitted to the training data which contains class labels representing red pixels. Using the labelled training data obtained using k-means, the KNN model was fitted to the training data. After which, the validation set was predicted for each pixel. A pixel could have been predicted to be in any class between 0 and 7. Classes 0 to 2 represent the red pixels, and classes 3 to 7 represent the background pixels.

E. Post Processing

After predicting the pixels in the validation and test data sets, it was determined for each pixel in the validation and test data sets whether or not it is a red vehicle. Pixels that were predicted to be in classes 0 to 2 were labelled as a red vehicle, and those predicted to be in classes 3 to 7, which is classified as background, were labelled as not a red vehicle.

III. EXPERIMENTS

A. Training and Validation

Similar to the 4-fold cross validation, a 70/30 train-test split was conducted on both the ground truth data and background data. After which, the 70% from the ground truth data and that from the background data were combined together, respectively, to form the training data set. Similarly, the 30% from the ground truth data and that from the background data were combined together, respectively, to form the validation data set. Using the KNN classifier from Scklearn, the model was fitted using the training data. After which, the validation data set was tested. An overall accuracy of 0.96828 was achieved. However, this is not representative of how well the model is able to identify red vehicles from the background. The high overall accuracy is likely due to the fact that the number of pixels for the background is much higher in comparison to that for red vehicles in the image. So when validating with both the ground truth and background, this imbalance is maintained; therefore, the predictions of the background data dominants the outcome of the accuracy. The model may not be fitted as well as it should be since the number of red vehicles are a minority within the training data set, which can be considered a bias within the algorithm. Therefore, the model was tested with only the validation data set from the ground truth, only pixels of the red vehicles. This achieved an accuracy of 0.653408, which indicates that model cannot accurately classify red vehicles in an image as well as it can the background. The confusion matrix shown in Table II indicates that the model can accurately predict when a pixel is not a red vehicle; however, the confusion matrix also indicates the model is more likely to incorrectly label a pixel which is not a red vehicle as being a red vehicle than to incorrectly label a pixel which is a red vehicle as being not a red vehicle. This results in a fall-out rate, which indicates the probability of detection, of 34.78%, which is can be considered somewhat high depending on the use for wanting to detect a red vehicle. For example, if the police department was looking for a suspect in a red vehicle, a fall-out rate over 30% may not be considered good when there is a limited amount of time in finding the suspect. The recall was 87.35%, meaning there is a high likelihood that the model will detect a red vehicle.

TABLE II
CONFUSION MATRIX

	Predicted	
Actual	Red Vehicle	Not Red Vehicle
Red Vehicle	5131	743
Not Red Vehicle	2688	109030

B. Testing

For testing, another image was given by Professor Zare. The coordinates chosen based on extracting the red pixels and detecting the red pixels that were corners, which was discussed in the Implementation section for pre-processing the test data, are shown in Figure 2. However, Figure 3 outlines in yellow some of the red vehicles that were not detected. This may have occurred because the pre-processing method for extracting the red pixels was unable to perceive those pixels as red if their RGB values were a different variation of red. In addition, it may have been unable to detect the red vehicles because the corner detection method failed to detect a corner pixel for those red pixels. After which, the RGB values of these chosen

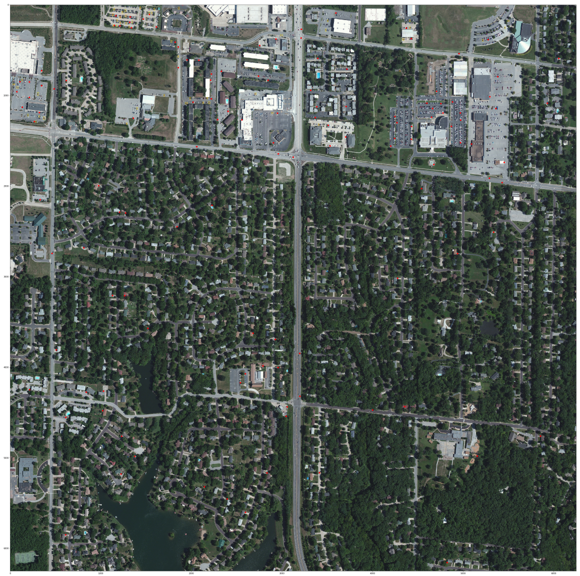

Fig. 2. Coordinates of Red Vehicles in Test Image

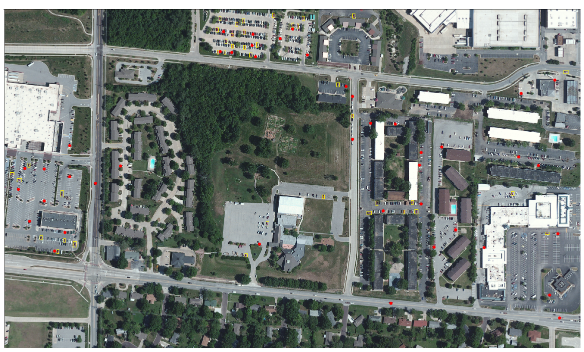

Fig. 3. Coordinates of Red Vehicles Not Detected in Test Image

pixels were tested with the KNN model that was created during the training and validation phase. The process of predicting whether or not these pixels were red vehicles is similar to the process in [1] by classifying each pixel as a red vehicle or not a red vehicle using the post processing method discussed earlier.

IV. Conclusion

While there are other optimal choices of algorithms for detecting vehicles in high resolution satellite images, using k-means and KNN still resulted in efficient results for the validation set for this project. Although the overall classification accuracy was over 90%, when predicting which class pixels in the validation set for only the ground truth, it was found that the classification accuracy decreased significantly, around 65%. While an accuracy of 65% is not horrible, it does indicate that there is room for improvement for detecting red vehicles in images. It would be beneficial to further investigate other methods for detecting a vehicle instead of relying solely on corner detection for indicating whether an object in the image is a vehicle. It is worth noting that although k-means worked well in pre-processing the training data in this experiment, it is possible that if the training data does not contain vehicle pixels that vary in RGB for the specified vehicle color, the KNN classifier may incorrectly predict which class the pixels in the test data belong. For future work, it would be worthwhile to investigate whether increasing the number of cluster centers for determining the variance in the red pixels and background pixels would increase classification accuracy when validating only the ground truth. Vehicle detection is an important problem that is vital to numerous domains. Therefore, it is necessary that we continue to put forth effort to improve or develop algorithms that will increase performance.

V. Notes

In the beginning, I discuss in general with Kiana Alikhademi about the project. All coding and work discussed in this paper are my own work and was completed by myself.

References

[1] H. Cheng, C. Weng and Y. Chen, *Vehicle Detection in Aerial Surveillance Using Dynamic Bayesian Networks*, in IEEE Transactions on Image Processing, vol. 21, no. 4, pp. 2152-2159, April 2012.

[2] H. Wei et al., *Vehicle detection from parking lot aerial images*, 2013 IEEE International Geoscience and Remote Sensing Symposium - IGARSS, Melbourne, VIC, 2013, pp. 4002-4005.

[3] X. Zheng and N. Liu, *Color recognition of clothes based on k-means and mean shift*, 2012 IEEE International Conference on Intelligent Control, Automatic Detection and High-End Equipment, Beijing, 2012, pp. 49-53.

[4] Bishop, C. (2006). Pattern recognition and machine learning. New York: Springer, pp.424-425.