COMPARISON OF MACHINE LEARNING ALGORITHMS TO DETERMINE TRAFFIC CONGESTION FROM CAMERA IMAGES

Corresponding Author:
Subhadipto Poddar
Institute for Transportation, Iowa State University
2711 South Loop Dr., Suite 4700, Ames, IA-50010
Tel: (515) 598-6807
Email: spoddar@iastate.edu

Koray Ozcan
Institute for Transportation, Iowa State University
2711 South Loop Dr., Suite 4700, Ames, IA-50010
Tel: (515) 708-0047
Email: koray6@iastate.edu

Pranamesh Chakraborty
Institute for Transportation, Iowa State University
2711 South Loop Dr., Suite 4700, Ames, IA-50010
Tel: (515) 357-9076
Email: pranames@iastate.edu

Vesal Ahsani
Institute for Transportation, Iowa State University
2711 South Loop Dr., Suite 4700, Ames, IA-50010
Tel: (515) 708-0860
Email: ahsaniv@iastate.edu

Anuj Sharma
Associate Professor, Department of Civil, Construction and Environmental Engineering, Iowa State University
2711 South Loop Dr., Suite 4700, Ames, IA-50010
Tel: (402) 472-6391
Email: anujs@iastate.edu

Soumik Sarkar
Assistant Professor, Mechanical Engineering, Iowa State University
207 Lab of Mechanics, Ames, IA – 50014
Tel: (515) 294-5212
Email: soumiks@iastate.edu

Date of submission: August 01, 2017
Word Count: 4471 + 5 figures + 6 tables = 7221

ABSTRACT

For a country like United States where drivers spend $1200 per year on traffic jams, congestion detection is a subject of prime importance. In the present study images are obtained from CCTV cameras installed at different locations in the state of Iowa and five state-of-the-art shallow algorithms are explored to identify congestion. Features are extracted from the images using the OpenCV packages (Ski-Thomasi, ORB, and findcontours) and structured edge toolbox. The images are then segregated into test and training sets followed by combination of features using the training set. The two best performing ensembles are found to be OS (ORB and Ski-Thomasi) and OSS (ORB, Ski-Thomasi and Structured Edge Toolbox). The algorithms (k-NN, random forest and SVM) with variable parameters are applied on the training set to tune up the parameters. After that all the five algorithms - Naïve Bayes, k-NN, decision tree, random forest and SVM are applied on the test set. The results show that SVM has the highest f1-score of 86.73%. A sensitivity analysis is computed using receiver operating characteristic curve under time of day and camera conditions. It is seen that presence of glare or other inhibitions in the camera during the daytime leads to some misclassification of the traffic state. The results are within 5% of that obtained by deep learning algorithms. To conclude, the paper provides an option for a consumer to invest 60$ on a shallow algorithm and achieve an accuracy of 86.73% or to take up expensive deep model to improve the accuracy.

Keywords: *Congestion detection, image processing, machine learning*

INTRODUCTION

Traffic congestion has been defined by US Department of Transportation (USDOT) as "one of the single largest threats" to the economic prosperity of the nation (1). As of 2016, traffic jams cost nearly $1200 per year for a driver in United States in terms of fuel and time (2). A major contributor to this congestion are traffic incidents. Schrank and Lomax (3) showed that implementation of improved incident management procedures in 272 out of 439 urban areas resulted in reduction of 143.3 million hours and $3.06 million of incident-related congestion. For a country with five of the top10 worst congested cities in world (2), traffic congestion detection and analysis has become the topic of prime importance.

Traffic state detection is conventionally done using point based sensors, which include microwave radars (4). Researchers have also developed mechanisms to detect congestion by comparing the measures from inductive loops across different locations, or between different times at one single location (5, 6). With the increasing usage of navigation based GPS devices, probe-based data have also come up for speed estimation of segments of road and hence thereby detect congestion (7). With the recent advancements in image processing techniques and the availability of these packages on the different platforms like Matlab, python (8, 9, 10) and so on, has improved the vision-based detection accuracy. Apart from the evolution of image processing, numerous supervised classification algorithms have come up in the recent years, many of which has already been used to identify the traffic state detection (11).

High traffic density means increased traffic on a road and is characterized by slower speeds, longer travel times, and increased vehicular queuing. Hosne-Al-Walid (12) proposed a technique to measure traffic density using Support Vector Machine (SVM) approach. It takes the image of current state of traffic on a road. Passing the image through the proposed system to detect traffic density and finally using SVM to classify data. Efficiency of their proposed method is 60%. Methods like Hidden Markov Models have also been used in the previous work where they have used a region of interest to find out traffic density (13). Density, an indirect measure of traffic congestion have also been identified using camera as a virtual loop detector (14). In this study the authors used an online-SVM to come up with an accuracy of 89.43% under different illumination conditions. In a similar type of work, an accuracy of 95% is achieved for different type of brightness and weather conditions during daytime (15). Here a boosted SVM is applied to achieve the result.

All these works have some limitation or the other in terms of the image illumination or camera positions they used. Generally, state Department of Transportation (DOT) installs cameras in freeways and arterials for surveillance tasks. They play important role in identifying incident detection. In 2016, approximately 22.5% (572) of the incidents have been detected using these cameras in the state of Iowa. These cameras are used by traffic incident managers (TIMs) who can zoom, tilt and pan the cameras as per their need. Hence, using cameras for congestion detection in reality involves the additional challenges due to their frequent movement which can alter the default calibrations. Thus, algorithms should not rely on the exact placement of cameras and should be able to accurately detect traffic condition for different placement scenarios.

With the introduction of graphics process unit (GPU), deep neural network has clearly gained an edge over the machine learning algorithms. However, when it comes to the cost of implementing, a machine learning algorithm can be implemented in a palm sized Raspberry pi2 which costs around 60$ whereas the cost of GPU is 500$ per stream. Considering the cost of implementation, this work aims to identify some of the best performing shallow supervised classification algorithms to detect traffic congestion states from traffic camera images irrespective of the location, orientation and weather condition.

The paper is broadly divided into four sections. The following section describes about the data and it processing technique. This is followed by the proposed approach being adopted to detect congestion. It is followed by a detailed results and discussion and finally the conclusion.

DATA PROCESSING

This section deals in an elaboration of the data description and the preprocessing of the same.

Data Description

The dataset is procured from two sources – camera images and sensors spread across the highways and the interstates of entire state of Iowa especially near the major cities of Des Moines, Sioux City, Council Bluffs, Cedar Rapids, Davenport and Iowa City.

Images from 99 CCTV cameras are extracted at 5 minutes interval from October 2016 to March 2017. Out of 4 million extracted images, 500,000 are discarded due to faultiness in their position and the remaining are used for further analysis.

The sensor data are available at 20secs interval from each lane. Sensors which are placed at 0.5 miles within the cameras are selected for the analysis. The data size of the same is nearly 500 MB/day and is available in a high performance computing system. To procure it and aggregate it at an interval of five minutes, a map-reduce code is constructed using Apache pig (16, 17).

The images so obtained have both the directions, of which the maximum occupancy is selected for classification. Now as suggested in (18) an occupancy of 20% or more is considered as a congested whereas an occupancy below that is considered an uncongested. Using this knowledge the images are segmented into two classes. However, being highways and freeways in Iowa, there is huge class imbalance between congested and uncongested images. Only 1200 images are found to be in congested condition. For this reason the rest of the work is carried out using 1200 images randomly from both the classes to prepare a balanced training set. Out of these 2400 images, 1000 images are kept aside for testing. Another 17 more images classified as congested are added from location outside the state of Iowa to see how the classification algorithm works. On inspection of these 1017 images 27 of them are found to be misclassified, probably due to sensor error, so they are relabeled. It is to be noted that the experiment is implemented using Windows 7 with Intel Core i7-4790 CPU with 8GB of RAM.

Data Preprocessing

The sensor data is aggregated to five minute interval and smoothened prior to further processing. Wavelet transform is used for the same to remove the noise and outliers. An example of wavelet transform for a day from a sensor on I-235 near Des Moines at Kep Way is shown in Figure 1 below. There are several families of wavelets such as Haar, Daubechies, Biorthogonal, Symlets, Coiflets, Morlet, Mexican Hat, Meyer, and so on. Daubechies extremal phase wavelets transform is used in this analysis. Daubechies family wavelets are also called “dbN” which N refers to the number of vanishing moments. The higher N, the longer the wavelet filter and smoother the wavelet. Based on the data, db2 with level 6 was used for smoothening curve-like filter which follows most of variations of original signal.

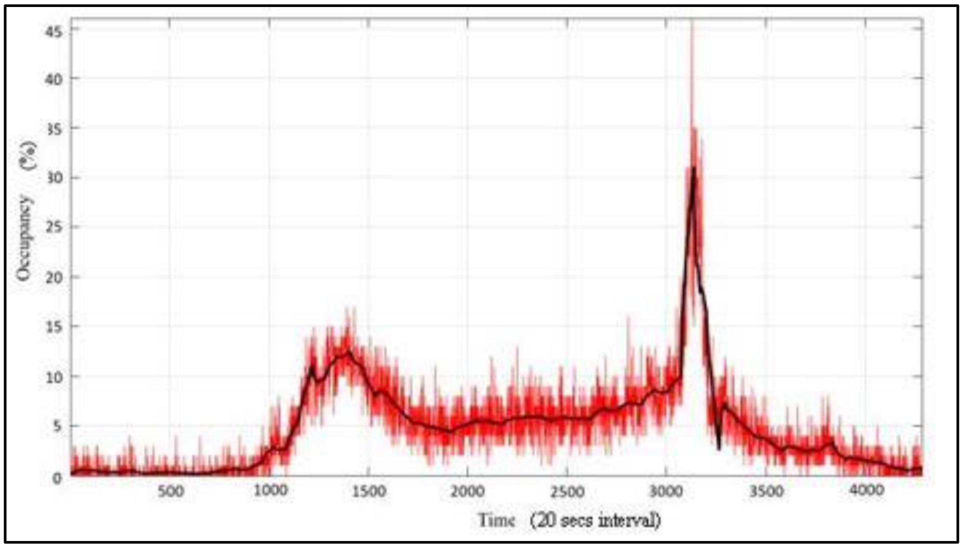


FIGURE 1 Smoothening of occupancy using db2 at level 6 method of wavelet transform.

PROPOSED METHODOLOGY

This section deals in mainly two parts - feature extraction which deals in the relevant algorithms used for extracting features and the supervised learning algorithms which use these extracted features to detect traffic congestion.

Feature Extraction

To extract the relevant features from the images, some of the popular inbuilt algorithms of OpenCV library in python and an edge detector toolbox implementation are explored. A brief description of each one of them is given below.

Ski-Thomasi Corner Detection

The Ski and Thomasi corner detection algorithm which identifies the corner points based on the size of the smaller eigenvalue of the tracking matrix (19). If this value is less than a certain threshold value, a point is determined as a corner point. An example of the implementation of this algorithm is shown in Figure 2(a).

Oriented FAST and Rotated BRIEF (ORB)

The ORB algorithm is another implementation developed in (10) to find out the corner points in an image. This algorithm extracts the keypoints using the FAST algorithm and then finds out the top N points amongst them using the Harris corner distance. Unlike the Harris corner distance alone, this has a modification which makes it robust to rotation. An example of this feature extraction is shown in Figure 2(b).

Find Contours

The findcontours algorithm of the OpenCV class extracts the location of change of contours of a binary image (18). An image consists of two types of connected component - l-pixels (l-component) and a connected component of O-pixels (background or hole). The algorithm detects all such 1-component and hole-component points of an image and returns the one which are called for. In this case all the contour points whether they are 1s or holes are summoned as it is assumed that the number of contour points will be related to the occupancy. An example of the contour points is shown in Figure 2(c).

Structured Edge Toolbox

The structured edge toolbox is a commercially licensed product available using the MATLAB image processing toolbox that proposes objects in an image (21). The presence of an object is determined as the probability of objectness that a bounding box to encompass the total number of edges wholly. The scoring function is a weighted sum of the edge strengths within a box minus those which are a part of the contour that grazes along its boundary. Objects or desired dimension can be detected by altering the objectness score and size of the sliding windows. An implementation of the same is shown in Figure 2 (d).

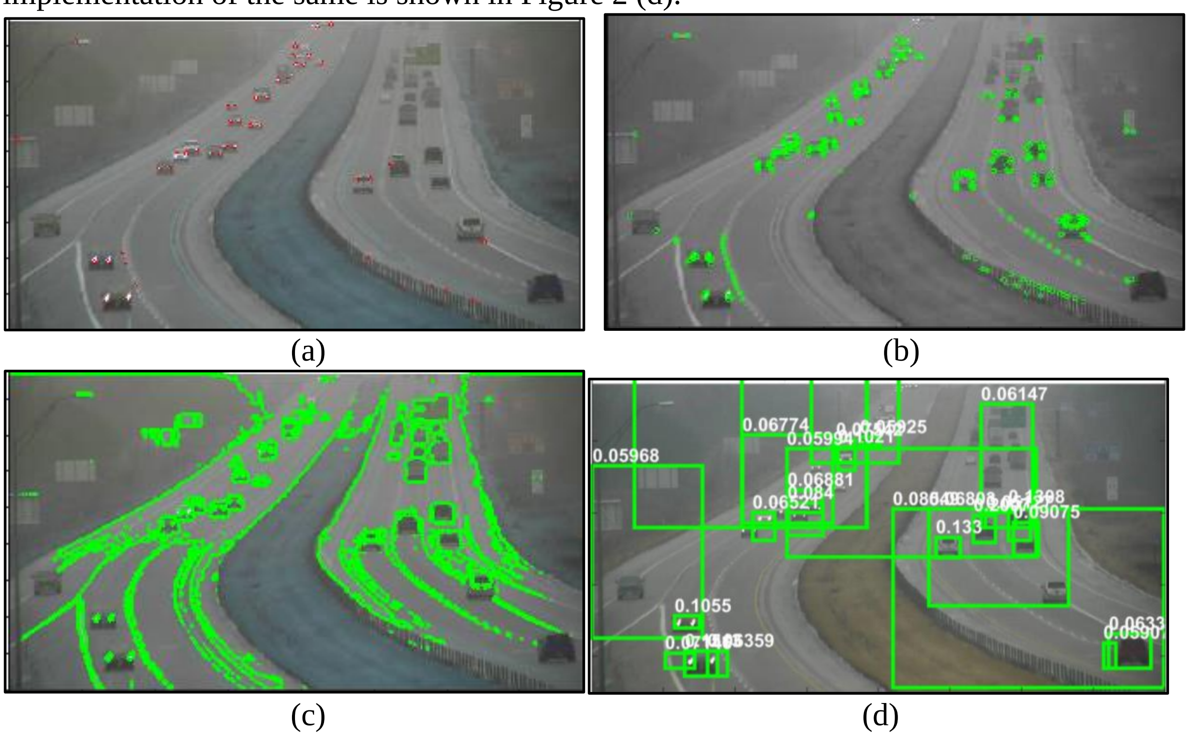


FIGURE 2 Camera image showing the a) corners detected by Ski-Thomasi algorithm, b) corners detected by ORB algorithm, c) corners detected by ORB algorithm, and d) objects detected by the Structured Edge Toolbox.

Supervised Classification Algorithms

Now after extraction of the features from the image, the next step is to use a suitable algorithm and classify the traffic state correctly. As the data to be used here is a labelled one, so the method used for classification is based upon the supervised learning algorithm. This section briefly describes the supervised algorithms that have been used for classification of traffic states.

Naïve Bayes

Naive-Bayes classification uses the Bayes theorem to classify and predict labels for the data. The assigning is based upon the maximum estimated posterior probability to the feature vector. It considers each feature to be independent of one another given its original class.

The main advantage of this method is that it requires a small amount of data for training purpose. However the major drawback is the low classification performance as compared to the other discriminative algorithms (22).

k-Nearest Neighbor

The k-Nearest Neighbor is a non-parametric method of classification which assigns label to a point in a feature space by considering its 'k' nearest neighbors. The nearest point in the feature vector space is defined by various metrics like Euclidean, cityblock, Manhattan, minowski and so on. Another variable feature is the number of nearest neighbor which can range from 2 to the sample size itself. The main advantages of kNN include rapid implementation and simplicity in using. The difficulty comes in determining the metrics as a small 'k' produces chaotic boundaries whereas a large k hides out the details.

Decision Tree Classifier

Another non-parametric method of machine learning that predicts the value (called leaves of a tree) from a set of features (called branches which lead to these leaves) is the Decision Tree classifier. The tree is learned by segregating the sources set into a subset of attribute set which is further divided into another subset in a recursive manner. The main advantage of this method is the visualization part. The tree structure can be easily interpreted. However, the main drawback is they often lead to overfitting.

Random Forest Classifier

The Random Forest Classifier is a method of ensemble learning which solves the problem of overfitting of the decision tree classifier. This method of classification starts with a random bagging of the features and then decision tree is applied to such bags. After accumulating all group of these decision trees, the random forest prediction is obtained. Apart from taking a little bit more time to run than the decision tree the random forest classifier works fairly well enough.

Support Vector Machines

Support vector machines categorizes data into different classes by solving a constrained quadratic optimization problem (23). The result so obtained is an optimal hyperplane. The 'optimal' hyperplane is determined by maximizing the largest minimum distance to the training examples as this one will be the one least sensitive to noise. The main parameter which the SVM depends on is the kernel size, varying which the accuracy can be altered. SVM is one of the most widely used shallow algorithms till date and has been often compared to the accuracy of the deep neural networks.

RESULTS AND DISCUSSION

The section is subdivided into three parts – feature selection and parameter determination, experimental results with analysis and the sensitivity analysis with the discussion.

Feature selection and parameter estimation

For evaluation of any algorithm used in this part and for the remaining analysis, the ratio of average to total f1-score, referred to as the f1-score, is used as the performance measure. The details are shown in equation 4. The f1-score is defined in equation 3 where the precision and the recall values are described in equations 1 and 2 respectively. If congestion is taken as the class to be determined, True Positive (TP) represents an image which was labeled congested and identified as congested, False Positive (FP) as an image which is labeled uncongested but considered as congested and False Negative (FN) as an image which is labeled as congested but is classified as uncongested.

$$recall = \frac{TP}{TP+FN} \tag{1}$$

$$precision = \frac{TP}{TP+FP} \tag{2}$$

$$f1-score = 2 * (precision * \frac{recall}{precision+recall}) \tag{3}$$

$$average\ to\ total\ f1-score = \frac{\sum_{i=1}^{2}(f1-score\ for\ class\ i * f1-score\ for\ support\ of\ class\ i)}{\sum Total\ images} \tag{4}$$

The training set is taken up and 10-fold cross-validation using standard Support Vector Machine with the penalty parameter of 1.0, kernel of 'rbf' is run to determine which feature combination works out best. SVM is chosen as it is seen that most of the works conducted before have used SVM for classification (16). First the algorithm is run on using a single feature as shown in Table 1. From this it is clear that the ORB feature gives the highest accuracy followed by structured edge toolbox. The findcontours gives the least accuracy so it is dropped for the rest of the analysis. The rest of the features are combined as shown in Table 2. From this it can be concluded that the combination of ORB, Ski-Thomasi works out to be the best followed by the ORB, Ski-Thomasi and structured edge toolbox. The rest of the analysis are done using these two feature vectors (OS and OSS) only.

TABLE 1 Average f1-score for SVM - one feature extractor at a time.

Base Feature	**Average f1-score**
Ski-Thomasi	78.28
ORB	**86.59**
findcontours	70.09
Structured edge detection toolbox	82.36

TABLE 2 Average f1-score for SVM – different combination of features.

Feature Combinations	**Average f1-score**
Ski-Thomasi, ORB	**88.28**
ORB, Structured edge detection toolbox	86.62
Ski-Thomasi, Structured edge detection toolbox	79.73
ORB, Ski-Thomasi, Structured edge detection toolbox	**88.08**

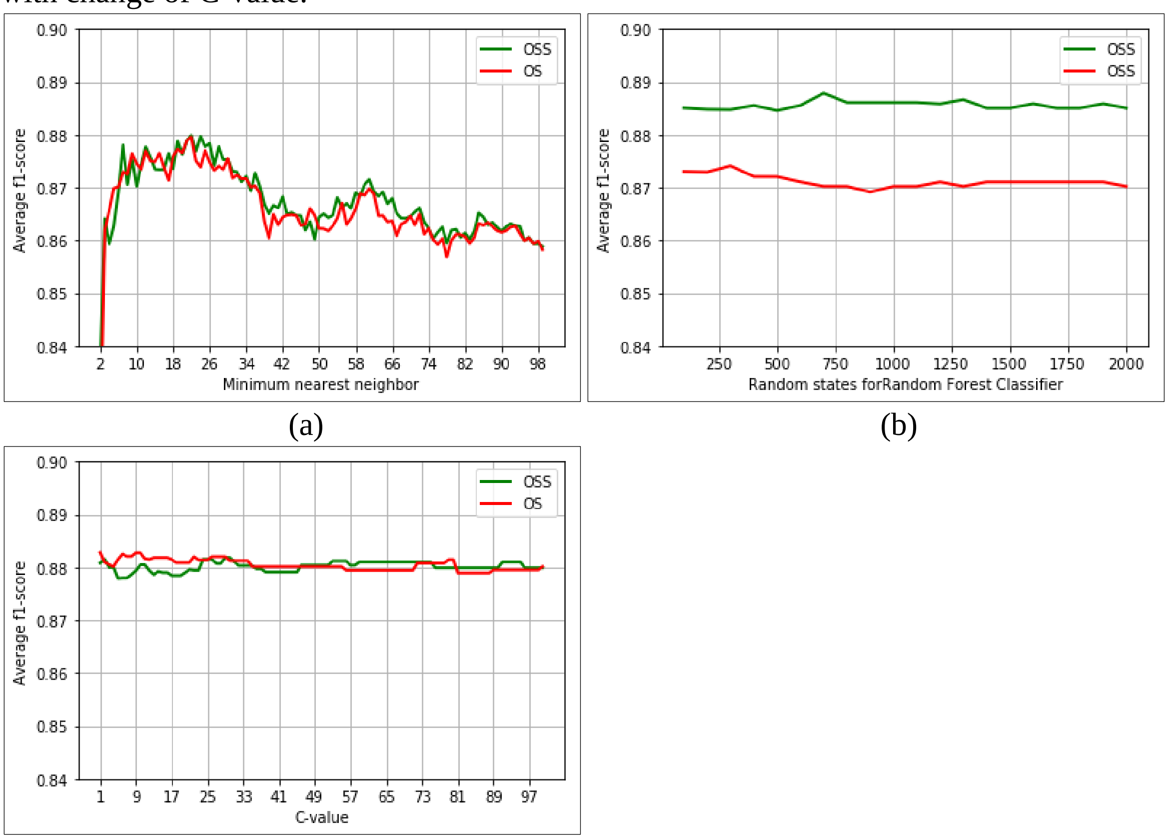
Experimental Results and Analysis

The feature vectors so selected are run on the rest of the algorithms and the parameters are tuned to optimize the maximum f1-score. 10-fold cross-validation is used for this purpose. The final model obtained from each algorithm is applied on the test set.

In case of kNN classification the number of nearest neighbor is changed in steps of 1 and the variation of average f1-score is shown in Figure 3(a). The maximum average f1-score obtained for OS and OSS are 87.96% and 87.98% respectively with 22 nearest neighbors.

For Random Forest Classifier the number of trees in the forest is the parameter of interest, so it is run for different combinations of trees as shown in Figure 3(b). The f1-score is hardly changing, however, the highest score achieved with for OS and OSS are 87.40% with 300 number of trees and 88.78% with 700 number of trees respectively.

For SVM there are two parameters which can be tuned – the C-value which optimizes the distance of the hyperplane from the data points and the kernel shape. First the kernel is varied for different types as shown in Table 3. This yields ‘rbf’ as the best performing one. This is followed by changing the C-value which maximizes the average f1-score at C=1 for OS and at C = 30 for OSS and as shown in Figure 3(c). It is also noticed that there is very little change in the f1-score with change of C-value.



(a) (b)

(c)

FIGURE 3 Variation of average f1-score with a) varying nearest neighbor values, b) varying number of number of trees for random forest classifier, and c) C-value for SVM

TABLE 3 Average f1-score for SVM for different kernel shapes.

Type of kerrnel	Average f1-score for OS	Average f1-score for OSS
Polynomial	87.18	86.93
Linear	86.97	85.73
RBF	**88.28**	**88.08**
Sigmoid	43.38	43.30

Now these models are run on the test set. The results of different algorithms are shown in Table 4. From this it can be concluded that the OSS performs better in all the cases as compared to OS. The highest f1-score of 86.73% is obtained using SVM followed by 86.53% by random forest.

TABLE 4 Average f1-score for different models on the test set images.

Model Algorithm	f1-score for OS	f1-score for OSS
Naïve Bayes Classifier	71.05	76.08
kNN Classifier	85.16	85.45
Decision Tree Classifier	82.21	82.70
Random Forest Classifier	84.47	86.53
Support Vector Machines	85.35	**86.73**

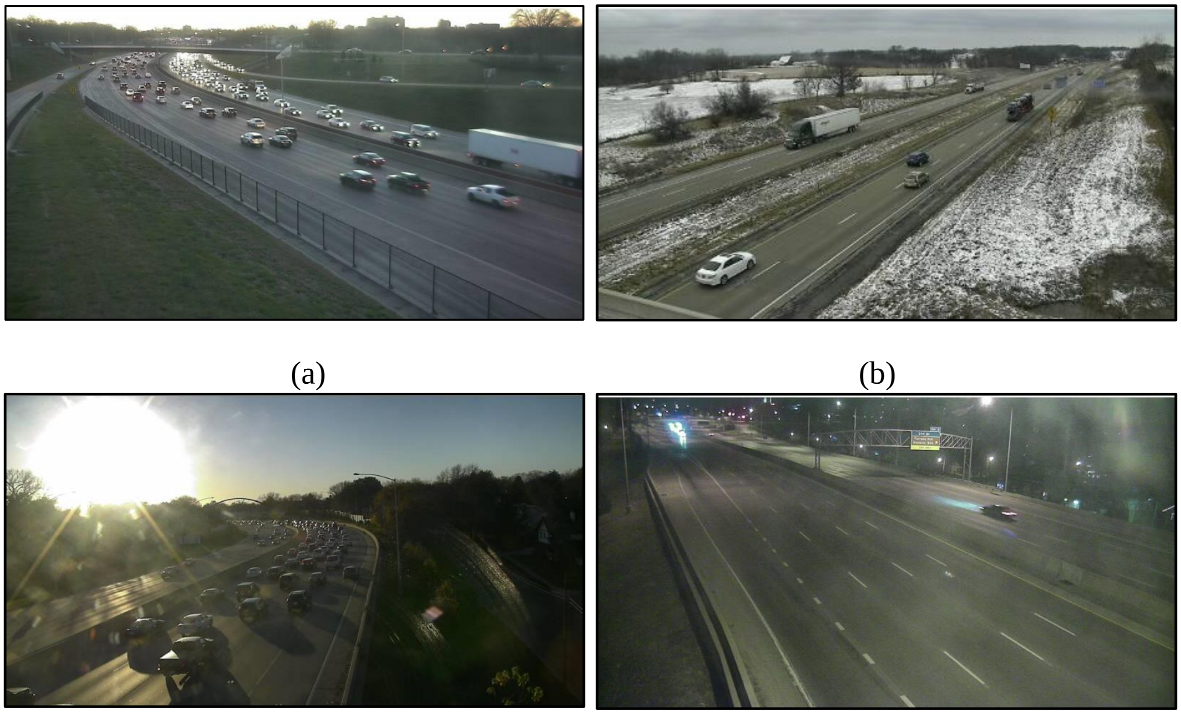
The result of the SVM is then taken up for evaluation of the performance metrics described here and for the sensitivity analysis after that. The details of precision, recall and f1-score is shown in Table 5 and confusion matrix is shown in Table 6. Figure 4 shows representative example of each group of classification. It can be seen from Figure 4 and rest of the FN images that the medium or low quantity of snow happens to be a major issue for an image to be classified as ‘congested’ whereas actually they are not. This is justified as the algorithm identifies them as corners in most cases. For FP images it is seen that the glare of the sun is a serious issue and images are most often misclassified as ‘uncongested’ state even when they are ‘congested’. In order to find out the cause of misclassification, sensitivity analysis is performed.

TABLE 5 Performance metrics for SVM using OSS.

Traffic State	Performance metrics f1-score for OSS			Support
	Precision	**Recall**	**f1-score**	
Congested	84.54	88.52	86.49	529
Uncongested	88.93	85.07	86.96	488
Average/total	86.83	86.73	86.73	

TABLE 6 Confusion matrix for SVM using OSS.

	Predicted Uncongested	Predicted Congested
Actual uncongested	450	79
Actual congested	56	432



(a) (b) (c) (d)

FIGURE 4 Examples of congestion detection (a) True Positive, (b) False Negative, (c) False Positive, and (d) True Negative

Sensitivity Analysis

This part describes the influence of the weather and environmental conditions and camera configurations on the congestion detection results. The sensitivity of the proposed system to two different conditions are assessed here. First one is the influence of the time of the day – day and night period and the second one is the camera resolution – presence of snow, rain, glare or blurred images (poor or good condition). To evaluate this system, the test dataset is divided into 4 subgroups according to the combination of the factors. These are:-

- Night under poor condition.
- Day under poor condition.
- Night under good condition.
- Day under good condition.

Receiver Operating Characteristics (ROC) curves are plotted for evaluating the performance of congestion detection. A ROC is a plot of True Positive rate with a False Positive rate (24). The True Positive and False Positive rates are described in equations 5 and 6 respectively where the TP, FN, FP and TN are all described before.

$$TPr = TP/(TP + FN) \tag{5}$$

$$FPr = FP/(FP + TN) \tag{6}$$

A good detection rate is based upon the area under the curve. For a poorly developed system, the true positive and the false positive rates become directly proportional to one another whereas for a well-developed system, the true positive rate is higher than the false positive one.

Figure 5 describes the true positive and the false positive rate for different classes. The area under the curves are also provided in each case. All the curves have the true positive rates higher than the false positive rates irrespective of the prevailing condition. Nights are seen to perform better than the day irrespective of the camera condition. Daytime with poor condition is the worst performing one which is quite understandable as there are quite a few of images with glare where the algorithm fails almost every other time. As the area under the curves is greater than 0.9 in most cases so the algorithm works great in most cases.

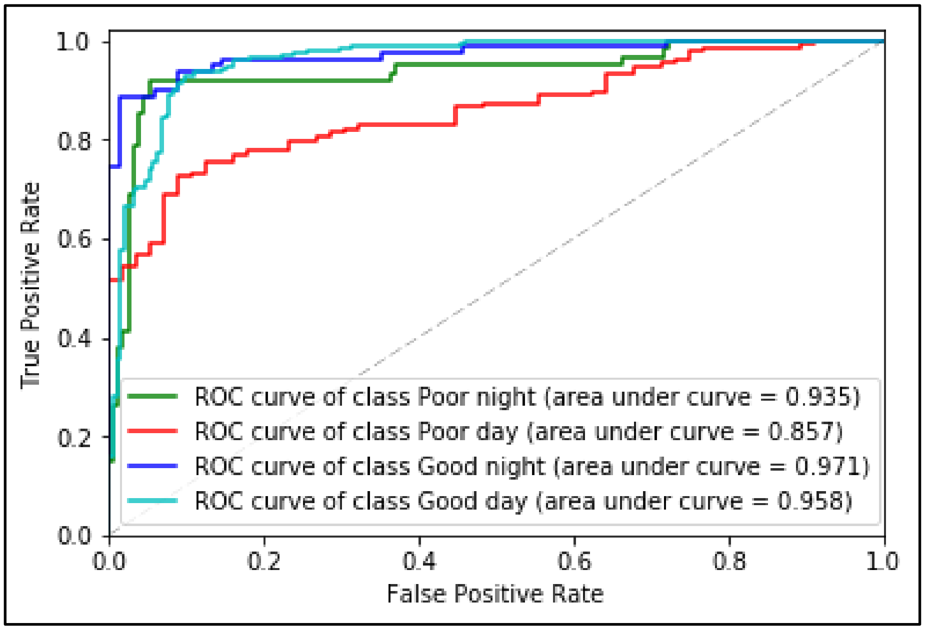


FIGURE 5 Receiver Operating Characteristic curves for different classes.

SUMMARY AND CONCLUSIONS

With the recent developments in image processing and machine learning algorithms, image analytics have improved to a great extent. This paper delivers a comparative study of five such state-of-the-art algorithms to come up with congestion detection irrespective of the camera location.

Traffic images are procured from CCTV camera images located all over the state of Iowa. They are then labelled as congested or uncongested using smoothened occupancy data collected from nearby sensors. Features are extracted from the images and algorithms are applied on them to find out that SVM works out to be the best one with an f1-score of 86.73% followed by random forest classifier with 86.53%. To find the reason for the misclassification, error analysis are performed. It is found that glare, rain, snow or blurred images during the morning time are the worst and greatly affects the efficiency of the algorithm. For other cases it gives a very high area under the curve value which shows that the algorithm is robust in most cases especially during the night time.

To conclude, this paper provides an option for the consumer where they can either invest a small amount, achieve a relatively good accuracy with a shallow model or they can spend almost 10 times more to carry out deep learning models for improving the accuracy.

ACKNOWLEDGMENT

Our research results are based upon work jointly supported by the National Science Foundation Partnerships for Innovation: Building Innovation Capacity (PFI:BIC) program under Grant No. 1632116, National Science Foundation under Grant No. CNS-1464279 and Iowa DOT Office of Traffic Operations Support Grant. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

REFERENCES

1. Owens, N., A. Armstrong, P. Sullivan, C. Mitchell, D. Newton, R. Brewster, and T. Trego, Traffic incident management handbook, 2010..
2. Traffic, I. accessed on Jan. 16, 2016.
3. Schrank, D. L., and T. J. Lomax. *2009 urban mobility report*. Publication Texas Transportation Institute, Texas A & M University, 2009.
4. Wang, Y., and M. Papageorgiou. Real-time freeway traffic state estimation based on extended Kalman filter: a general approach. *Transportation Research Part B: Methodological*, *39*(2), 2005, 141-167.
5. Courage, K. G., and M. Levin. *A freeway corridor surveillance, information and control system*. Publication Texas Transportation Institute, Texas A & M University, 1968.
6. West, J. T.. California makes its move. *Traffic Engineering*, *41*(4), 1971, 12-18.
7. Dailey, D. J., F. W. Cathey, and S. Pumrin. An algorithm to estimate mean traffic speed using uncalibrated cameras. *IEEE Transactions on Intelligent Transportation Systems*, *1*(2), 2000, 98-107.
8. Bay, H., T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. *Computer vision–ECCV 2006*, 2006, 404-417.
9. Lowe, D. G.. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, *60*(2), 2004, 91-110.
10. Rublee, E., V. Rabaud, K. Konolige, and G. Bradski. ORB: An efficient alternative to SIFT or SURF. In *Computer Vision (ICCV), 2011 IEEE international conference on* (pp. 2564-2571), 2011, IEEE.
11. Yuan, F., and R. L. Cheu. Incident detection using support vector machines. *Transportation Research Part C: Emerging Technologies*, *11*(3), 2003, 309-328.
12. Hosne-Al-Walid, , Traffic Density Measurement using Image Processing: An SVM approach. *International Journal of Innovative Research in Computer and Communication Engineering*, Vol. 3, Issue 6, 2015.
13. Ishak, S., and H. Al-Deek. Performance of automatic ANN-based incident detection on freeways. *Journal of Transportation Engineering*, *125*(4), 1999, 281-290.
14. Wassantachat, T., Z. Li, J. Chen, Y. Wang, and E. Tan. Traffic density estimation with on-line SVM classifier. In *Advanced Video and Signal Based Surveillance, 2009. AVSS'09. Sixth IEEE International Conference*. 2009. (pp. 13-18). IEEE.
15. Li, Z., E. Tan, J. Chen, and T. Wassantachat. On traffic density estimation with a boosted SVM classifier. In *Computing: Techniques and Applications, 2008. DICTA'08. Digital Image*. 2008. (pp. 117-123). IEEE.
16. Addisalem, M., K.P. Porkodi, and I. Karthika. Smart cities using big data analytics: a way ahead for world. *EPRA International Journal of Research and Development*, Volume 2, Issue 4. 2017.

17. Olston, C., B. Reed, U. Srivastava, R. Kumar, and A. Tomkins. Pig latin: a not-so-foreign language for data processing. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data* (pp. 1099-1110), 2008, ACM.

18. Persaud, B. N., and F. L. Hall. Catastrophe theory and patterns in 30-second freeway traffic data—Implications for incident detection. *Transportation Research Part A: General*, *23*(2), 1989, 103-113.

19. Tomasi, C., and T. K. Detection. *Tracking of Point Features*. Techical Report CMU-CS-91-132, Carnegie Mellon University, 1991.

20. Suzuki, S. Topological structural analysis of digitized binary images by border following. *Computer vision, graphics, and image processing*, *30*(1), 1985, 32-46.

21. Zitnick, C. L., and P. Dollár. Edge boxes: Locating object proposals from edges. In *European Conference on Computer Vision*, 2014, pp. 391-405. Springer, Cham.

22. Khan, A., B. Baharudin, L. H. Lee, and K. Khan. A review of machine learning algorithms for text-documents classification. *Journal of advances in information technology*, *1*(1), 2010, 4-20.

23. Dreiseitl, S., and L. Ohno-Machado. Logistic regression and artificial neural network classification models: a methodology review. *Journal of biomedical informatics*, *35*(5), 2002, 352-359.

24. Hanley, J. A., and B. J. McNeil. The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology*, *143*(1), 1982, 29-36.