Developing Traffic Congestion Detection Model Using Deep Learning Approach: A Case Study of Addis Ababa City Road

Bedada Bekele Dursa* and Kula Kekeba Tune

Abstract

The traffic system is one of the core requirements of a civilized world and the development of the country depends on it in many aspects. In Ethiopia, the number of vehicles and pedestrians is increasing at a high rate from time to time. Excessive numbers of traffic on roads and improper control of traffic create traffic congestion. Uncontrolled traffic congestion hinders the transportation of goods and commuters from place to place and increases the volume of carbon emitted into the air. It can also either hamper or stagnates schedule, business, and commerce. Many images and video processing approaches have been researched in the literature on how to detect traffic congestion. One such approach is that of using background and foreground subtraction, convolutional neural network, and Average frame difference and deep learning method used to detect traffic congestion from different video sources. From the review one-stage object detector identified as the best methods to detect traffic congestion with acceptable accuracy and speed. In this study one-stage object detectors are used to detect traffic congestion from recorded video. Data is collected from different video footage and frames extracted from videos to prepare a dataset for the thesis. The extracted frames were labeled manually as congested and uncongested. To train, the models pre-trained weights were used. YOLOV3 and YOLOV5 model used for experimentation. Accuracy and speed metrics used to evaluate the performance of the models. A YOLOV3 model achieved 41.6 FPS and 68.6 % mAP on a testing dataset.

Keywords: Traffic Congestion Detection; YOLOV3; YOLOV5; Deep Learning; Darknet53; Convolutional Neural Network

Introduction

Traffic congestion is an unavoidable consequence of scarce transport facilities such as road lane, parking area, road signals, and effective traffic management. Traffic congestion can also be perceived as slowed traffic flow below reasonable speed as a result of the unbalance between the number of vehicles trying to use a road and the road capacity of the traffic network [1]. An intelligent transportation system is advanced technologies that include computer vision, information processing, communications, and control system to improve the efficiency and capacity of roads to reduce traffic congestion [2].

Traffic congestion detection (TCD) becomes one of the most important issues in the smart transportation system. Since the traffic system is the most important requirement for urban city day- to -day activities, awareness of traffic congestion...
is crucial to select routes within low traffic flow to reach the destination in a short
time and with minimal cost [3]. Traffic congestion is unavoidable especially in the
city where the number of vehicles increases at a high rate and traffic management
lacks modern technologies to control road traffic. Most of the time traffic congestion
has occurred as a result of road lane occupied by large numbers of vehicles, traffic
incidents, work zone, bad weather, poor signal timing, and special events [4]. The
Cause of traffic congestion, location, and the number of congestions in Addis Ababa
presented in Table 1.1 according to the study conducted by Hager Yilma.

Traffic Congestion has many effects on the life of a society in both developed and
developing countries. Traffic congestion diminishes the productivities and increases
the overall cost of transportation services for the freight industry and trucking com-
panies [4]. It also reduces the number of business labor market, customer delivery
market, and shopper market areas, that can be accessed within a limited window
of reasonable travel time. This in return affects the businesses by reducing their ac-
cess to specialize material inputs and labor by reducing the scale of their customer
markets[1]. The volume of greenhouse gases emitted to air and fuel consumption
increases with stop-and-go driving caused by traffic congestion[5].

Various sources estimated the current population size of Addis Ababa city to be
3 to 5 million, although the official statistics of 2008 census put its population size
to 2,112,737 [6] and the total number of registered vehicles reached 630,440 until
30/10/2012 E.C according to the official Facebook page of the Ethiopian Ministry
of Transport[7]. From the total number of vehicles, 52.5 % of the vehicles were
found in Addis Ababa, and the number of vehicles grew by 5% yearly. Urbanization
is accompanied by a high population and the number of vehicle growth is usually
accompanied by high traffic congestion [8]. Within such numbers of population and
vehicle traffic congestion is inevitable in Addis Ababa city. During peak hours in
Addis Ababa very large queues of people noticed and the speed with which vehicles
travel is slow.

The road transportation system is at the heart of the development of Ethiopia in
many aspects. Addis Ababa connected with all regions to exchange an agricultural
product, commercially goods, and imported-export goods. This leads to high traffic
congestion occurrence every day. Addis Ababa is also connected to international
countries such as Djibouti, Eritrea, Kenya, and Sudan. According to a survey con-
ducted by Hagere Yilma (2014), on a commuter who travels through route from
Kolfe18 to Autobis Tera and Arakillo to Piazza and Merkato responded that 68%
of them experience traffic congestion daily. While 20 % encountered congestion 2-3
days in a week and 6 % of them experience congestion once a week [9]. This indi-
cates that there is a high traffic congestion problem in Addis Ababa city.

So, the consequence of traffic congestion limits the benefits that society and gov-
ernment should get from road transportation. Traffic congestion also has a direct
impact on the life of our country communities. Extra fuel, time, accident, and pay-
ing extra money are a few consequences of traffic congestion that the community
suffers in their daily life. Therefore, developing a model that can detect traffic con-
gestion from the surveillance camera can reduce the number of traffic congestion
that occurred across the country.
Table 1 . Cause of Traffic Congestion in Addis Ababa City [9]

| Sub City          | Number of Congested Areas | Cause of Congestion                                                                 |
|-------------------|---------------------------|-------------------------------------------------------------------------------------|
| Akaki Kality      | 3                         | Narrow lane.                                                                        |
| Nifas Silk Lafto  | 7                         | Narrow lane, roadside parking.                                                      |
| Yeka              | 3                         | Unavailability of the alternative lane, roadside parking.                            |
| Kolfe Keranyo     | 3                         | Unavailability of the alternative lane.                                             |
| Lideta            | 4                         | Road construction and development work.                                              |
| Kirkos            | 8                         | Operators not using the available alternative roads.                                |
| Gullele           | 12                        | Poor road condition, unavailability of traffic signals, no sidewalk for pedestrian, and roundabout. |
| Addis Ketema      | 3                         | Narrow road and roundabout.                                                         |
| Arada             | 8                         | Roadside parking.                                                                  |
| Bole              | 4                         | Heavy truck, too many vehicles at taxi terminals.                                   |

Contributions

In this study, a traffic congestion detection model was proposed which is capable of detecting both recurrent and non-recurrent traffic congestion from video sources with promising accuracy and good speed. The main contribution of the work is to prove that traffic congestion can be detected from a surveillance camera in Addis Ababa city. For experimentation purpose dataset prepared from different video footage. The prepared dataset can be used as an initial benchmark dataset in the domain of traffic congestion in the case of Ethiopia. Additionally, this thesis is the first research in the domain of traffic congestion detection from surveillance cameras in the case of Ethiopia. So, the work can be used as a starting point by other researchers to solve different problems in the domain of traffic congestion.

Literature Review

This section describes the related work that related to the domain of the work. The traffic congestion detection techniques, one-stage object detector and related work included.

Traffic Congestion Detection Techniques

Currently, in the field of computer vision, different techniques were used to detect traffic congestion. Puntavungkour et al. detected traffic congestion using high-resolution aerial image sequences using Haar-like features and Adaboost techniques [10]. They also used a support vector machine (SVM) for classification and clustering pixels of images. Image processing based on surveillance camera works much better than another technique because it functions by visualizing a vehicle in a video. And it is budget-friendly compared to other detecting techniques. However, it is affected by different environmental factors such as illumination, lighting, and bad weather conditions[11].

Liepins et al. also proposed a solution for TCD using magnetic sensors[12]. In their work non-invasive wireless magnet sensor networks used to count vehicles for intelligent traffic management systems. The proposed method gives stable and robust results because magnetic sensors are not affected by weather conditions. Gholve et al. used embedded wireless magnetic sensors to detect congestion[13]. They developed a prototype to communicate a node of the sensor to traffic congestion between anodes. Barbagli et al. used Acoustic sensors to collect real-time traffic data from
the motorway by deploying a sensor on the side-way of the road [14]. The proposed method is capable to detect complete real-time traffic information at a specific time.

Boris et al. proposed a traffic congestion analysis method based on a video sequence collected from an optical sensor installed on each lane of the road [15]. They divide each sensor into two zones (entry and exit) which they used to identify the direction of vehicles and estimate the speed of vehicles using the distance between zones. A deep learning approach [16] is used currently in the field of computer vision to detect traffic congestion.

**One-stage Object Detector**

In the world of computer vision nowadays, there are two main types of object detectors, namely, one-stage object detector and two-stage object detector[17]. Region proposal-based detector is another name for a two-stage object detector. Region-based Convolutional Neural Network (R-CNN), Spatial Pyramid Pooling (SPP-net), Fast Region-based Convolutional Neural Network (Fast-RCNN), Faster Region-based Convolutional Neural Network (Faster-RCNN), and Region-based Fully Convolutional Network (R-FCN) are some of the algorithms which categorized as two-stage object detector.

Two-stage object detectors generates a region of interest using the region proposal network in the first stage and stage two object classification and bounding-box regression performed using the region proposals generated in the first stage. Such kinds of models are slower and accurate compared to the one-stage object detector. On the other hand, there is a one-stage object detector such as You Only Look Once (YOLO), RetinaNet, and Single Shot multibox Detector (SSD) categorized under one-stage object detector which is also known as an end-to-end strategy[18].

One-stage object detector accepts an input image and learns the bounding box coordinates and class probabilities treating object detection as a simple regression problem[17]. These types of detectors usually target reasonable accuracy with high processing speed. The main reason behind low accuracy compared to two-stage object detectors is that the candidate proposal extracted by one-stage object detectors cause extreme class imbalance during training because it contains too many well-classified backgrounds (10^4 - 10^5) [19]. Recently one-stage detectors have received high attention from researchers because of their higher speed performance. In the year 2016 Liu et al. proposed SSD which become the baseline of most newly proposed one-stage object detectors[20].

One-stage object detectors first generate some low-level feature maps using backbone models and then extract high-level feature maps more semantic information by adding several consecutive convolutional layers[21]. One-stage object detectors use a focal loss to improve detection accuracy. Focal loss adjusts the weights of the conventional losses by utilizing the probabilities that derived in each forward propagation. This enables focal loss to down-weight the predominant easy negatives by automatically changing weights during the training of the networks of one-stage object detectors [19].

**Related Work**

Lam et al. used images provided by a local government from 38 different locations to detect traffic congestion [22]. Their work contains three main parts, in the first
part, on-line real-time image provided by government website downloaded and stored on their storage. Then as the second part, they used traffic signs on road and Haar-like features to detect and count vehicles. In the third part, congestion is detected using a correlation coefficient and stored in a database where government agencies and road users extract information using mobile apps and websites.

Wang et al. proposed methods to detect traffic congestion using classifiers built based on CNN [16]. In the approach, they used SVM after the architecture of CNN to classify congestion and non-congestion state of traffic. They used transfer learning to train and test their model. They prepared a dataset consists of 30000 congested and 20000 non-congested images from different surveillance video of 24 hours. To classify congested and non-congested traffic state four typical architecture of CNN based on AlexNet and VGGNet using transfer learning. They manually labeled 100 images for four levels (no vehicles, small traffic density, large traffic density, and congested traffic). To use transfer learning, they removed the last three fully connected layers of AlexNet and VGGNet so that they included new layers to classify the congested and non-congested state of traffic. Their model achieved 90% accuracy according to their experiment result.

M. Sujatha et al. proposed a system to monitor real-time traffic congestion and notify the waiting time to the public on their mobile phone [23]. In the work, they used a static camera to get input video for their system. Canny edge detection algorithm and background subtraction used as the main algorithms to detect congestion. To get the number of vehicles found on a road at a specific area, an area covered by two-wheeler and four-wheeler calculated. Then the total white pixel was calculated and divided by the area of the two-wheeler individually.

Mohammad et al. proposed area-based real-time traffic congestion detection approach using image processing to control traffic light automatically [24]. The technique performs well using a simple algorithm to calculate vehicle density using an area of a road occupied by the edge of the vehicle. However, when a large homogenous surface of the vehicle appears on camera their technique produces low accuracy.

**Methodology**

This section describes the preparation of data set that used for training, validation, and testing the models. Frame extraction, data labeling, and data set partitioning included.

**Data set Preparation**

The authors have used a video recorded by the Addis Ababa police commission and from three different online repositories. Both local and non-local video clips are collected from shutterstock.com, pond5.com, and gettyimages.com. The video data set contains video from main road traffic recorded during day hours. The collected video content includes traffic video footage that has congested and uncongested state, from different routes in the city, and which recorded during sunny, rainy, and cloudy times.

The authors have extracted frames from surveillance videos to prepare the training and testing data set needed for the experiments. This is because video can
comprise 20 to 30 frames for each second [25]. One frame per second extracted from the video clips and resized to 500 x 500 pixels. The authors finally managed to collect 4234 frames in total with congested (2183) and uncongested (2051) traffic scenes from both local and non-local video sources.

Bounding box used to indicate the region of interest while labeling images. An image is labeled manually to generate the annotation for each image. From different annotation types bounding box is used because it uses a rectangular box to define the location of the targeted object. This technique of defining a location of a targeted object is more suitable for YOLO than other annotation types. Total images data set split into two using a 90:10 ratio of training (3810) and testing (424) data set. Then again using a 90:10 ratio the training dataset split into training (3429) and validation (381) data sets. Figure 1 depicts the overall data partitioning process.

![Figure 1 Dataset Partitioning](image)

**System Architecture**

YOLOV3 split an image into S x S grid to predict a fixed number of boundary boxes for each grid cell to detect congested and uncongested traffic scenes from the input image. There are 9 pre-selected clusters of anchors distributed equally across three scales and each group is assigned to a specific feature map to train the model. To train the model images in the training data set iteratively passed to the network of the model and validation data set used to access the performance of model during training in terms of mAP, precision, and recall to check how the model is learning. After accessing the performance of the model, the weight of the model that gives the best performance selected and saved for detecting traffic congestion. Then the test phase is performed using images from the test data set which are unseen by the model during training phases. Finally, the model predicts the confidence score and predicts the probability of congested and uncongested traffic scenes within the given image. Figure 2 highlights the overall process of YOLOV3 model training and testing on the data set to detect congested and uncongested traffic scene.

![Figure 2 System Architecture](image)

Input image normalized by dividing each pixel value by 255 to reduce the variance in the values for each pixel because image pixel values fluctuate between 0 and 1. Then the normalized image is divided into a grid, based on a predetermined level of graininess. Then the normalized image is accepted by Darknet53 to start the process of feature maps extraction from the object of interest. The prediction of the object is performed at three different scales. In this research image with a dimension of 416 x 416 is used during training. And to get the first prediction many convolutional layers perform on the input image with the stride of 32 to produce a feature map.
of 13 x 13, and then 7 times 1 x1 and 3 x3 convolutional kernels are processed to realize the first class and regression bounding box.

The second prediction achieved by processing a feature map of 13 x 13 dimensions 5 times by 1 x1 and 3 x 3 convolution kernels, followed by 2 times the upsampling layer, and a feature map of 26 x26 obtained using a stride of 16. And the new dimension is used to achieve the second category and regression bounding box by processing 7 times using 1 x 1 and 3 x3 convolution kernels. The last prediction was achieved by applying 1 x 1 and 3 x 3 convolution kernels on 26 x 26 feature map 5 times, and by upsampling two times to get 52 x 52 feature maps. Then, 7 times 1 x 1 and 3 x3 convolution kernels applied to the new feature map to predict the third category and regression bounding box. Finally, the model produces (13 x 13) + (26 x26) + (52 x 52) x 3 bounding boxes to detect congested and uncongested traffic scene. The model performs two operations to minimize 10,647 bounding boxes to a single box.

First, the model filters the boxes based on the objectness score. To filter boxes the model removes the bounding boxes that have an objectness score less than the threshold value 0.7. Then, the second operation is performed to removes the remaining bounding boxes based on IOU. This operation is called NMS and intended to remove multiple boxes that detect the same image. Boxes with the highest overlap will be suppressed. IOU is an overlap of the predicted (ground-truth) box area with another bounding box.

**Experimentation**

To detect traffic congestion from the recorded video a one-stage object detectors is used. Frames with congested and uncongested traffic scenes extracted from video to be used as training, validation, and testing data sets in all experiments. Different experiments are conducted to evaluate the detection and speed performance using YOLOV3 and YOLOV5 models.

To start YOLOV3 model training the prepared data set is compressed and uploaded to Google Drive. Then the data set extracted into a directory called data. For YOLOV3 data set split is performed by creating a .txt file that contains the path to images. So, three text files (train.txt, valid.txt, and test.txt) were created for training, validating, and testing the model. Additionally, three configuration files need to start the training phase. The first file is named as obj.data which specify the number of class and paths to training and validation data set during the training phase. This configuration file also specifies the testing data set path during the testing phase.

The second configuration file is given a file name called obj.names, this file contains a list of classes (congested and uncongested) which model compare with the labels created during the data preprocessing. The third configuration file name after the model name (yolov3.cfg). This configuration file contains all hyperparameters that fine-tuned to select the best model. Once necessary files are adjusted weight for initiations of model downloaded into weights directors and training of the model starts. During the training process, two weights of the model were saved with best.pt and last.pt file names. To test the performance of the model the saved weights are used.
To start YOLOV5 model training different configuration and data partitioning performed. For this model images and labels of the images are separated and stored in a directory called images and labels. Images and labels under images and labels directory, split into training, validation, and testing data set and store under a train, valid and a test directory. Once a split of the data set is over, images and labels directory stored under a directory called data. To upload a data set into Google Drive compressing the data directory is need to speed up the uploading process. To use an uploaded compressed data set on Colab unzipping the data set is needed. So, the data set is extracted into a training directory.

To use freely available GPU on Colab, before connecting the Colab notebook to Google Drive changing runtime types into GPU is must. Google provides one of the available GPU types randomly. But to get Tesla P100-PCIE GPU with 16 GB memory it is necessary to execute factory reset runtime and check if the types that need is provided. YOLOV5 needs two configuration files to be adjusted to start the training process. The first configuration file name is Data set.yaml, within this file class number (2) and class labels (congested and uncongested) are identified. Additionally, a path to train, valid, and test data set specified in this file. The filename of the second configuration file is yolov5.yaml. All hyperparameter that used for fine-tuning to select the best model is stored under this configuration file.

**Result and Discussion**

Figure 3 depicts the performance of YOLOV3 model on the validation dataset during a training phase. As indicated on the plot the model train for 50 epochs. The evaluation metrics such as precision, mAP and recall graph show the model ability to detect traffic congestion from the given data set. The evaluation metrics used is included below begin equation

\[
Precision = \frac{TP}{TP + FP}
\]  

(1)

\[
Recall = \frac{TP}{TP + FN}
\]  

(2)

\[
mAP = \frac{\sum_{i=1}^{k} AP_i}{k}
\]  

(3)

**Figure 3 YOLOV3 Accuracy Metrics on Validation Data set**

Figure 4 presents the accuracy metrics comparison of validation and test dataset for YOLOV3 model. The model achieved the same precision for both datasets. However, the model performance shows a small difference in terms of recall and mAP metrics. The model performance indicates that there is 2.6 % gap between validation and test datasets while evaluating the detection performance using mAP. The recall metrics also indicate that the model performance improved by 3 % on the validation dataset than on the testing dataset. In general, the performance of the model is good because the gap of accuracy metrics on validation and test dataset is not high.
Figure 5 depicts YOLOV5 model performance on the validation dataset during the training phase. The precision of the model fluctuates between high and low value during the training process with a small gap, except at the first epoch when the value rises to 0.5. The recall value of the model increases with a large gap for the first 20 epochs and fluctuates with a small gap for the rest of the epochs. A classification value dropped at a high rate during the first 10 epochs and fluctuates at a low rate for the next 15 epochs. And remain almost the same for the last 25 epochs. The model mAP increases starting from the first epochs toward the last epochs. But it dropped with a small value for few epochs.

Figure 6 depicts a comparison of accuracy metrics for validation and testing dataset for YOLOV5. The model performs with a small precision difference on a validation and testing dataset which is 2.1 %. However, the difference increases to 3.1 for recall and 7.1 % for mAP. In general, the performance of the model on the validation dataset is good compared to the testing dataset. This indicates that the model gets chances to learn the validation dataset during training because the validation dataset is used many times during training to select the best model.

Comparison of YOLOV3 and YOLOV5 Results
YOLOV3 and YOLOV5 accuracy and speed performance on the test dataset is compared in this section. Figure 7 indicates that YOLOV3 outperforms YOLOV5 in terms of the presented accuracy metrics. The models achieved almost similar recall value with less than 1 % difference between the models. But performance difference increases with precision to 6.3 % with mAP. This indicates YOLOV3 can detect traffic congestion 6.3 times accurately than YOLOV5. The performance of the two models evaluated with precision metrics and the result indicates that YOLOV3 detects precisely 8.1 times than YOLOV5. Figure 8 shows the speed performance of the two models. From the graph, it is clear that YOLOV5 outperforms YOLOV3. YOLOV5 achieved 61.6 FPS on video and the model took 17.5 seconds to detect congested and uncongested traffic scenes from the test dataset which contains 424 images. YOLOV3 took 21.9 seconds to complete detection on the same test dataset and process video with 41.6 FPS.

YOLOV3 achieved good accuracy on test datasets than YOLOV5 but speed metrics indicate that YOLOV5 is good in terms of speed. In most deep learning models,
there is a tradeoff between accuracy and speed. So, based on performance evaluation performed on the test dataset it proves that YOLOV3 outperforms YOLOV5 by detecting traffic congestion accurately. While YOLOV5 outperforms YOLOV3 by processing speed. The authors of the thesis suggest that YOLOV3 is good to detect traffic congestion from surveillance cameras videos. The main reason behind suggesting YOLOV3 is the experiments result in terms of accuracy. And video can comprise 20 to 30 frames for each second [25]. This indicates that YOLOV3 performance to process video which is 41.6 FPS is enough to process video.

Conclusions
Traffic congestion has become a pressing concern for both developing and developed countries. Especially a country like Ethiopia whose socio-economic dependency is high on-road transportation needs to have a technique to detect traffic congestion from a surveillance camera. Traffic flow in cities like Addis Ababa should be optimal enough that there should be minimum congestion on roads and roads should be utilized fully. Uncontrolled traffic congestion has many consequences, but the most important one is the increased emission of greenhouse gases. Traffic congestion increases the wastage of fuel, time, and cost of transportation which in return makes the commuters feel stressed especially during the peak hours of the day. Therefore, developing a model that can automatically detect traffic congestion from video using deep learning techniques reduces the problem caused by traffic congestion in everyday activities.

In this research, a one-stage object detector model has been presented and proposed to detect both congested and uncongested traffic scenes from recorded video. The proposed model can be used as a tool to detect traffic congestion from a surveillance camera and it can be used as input to automatic traffic light controlling system. To experiment with the models PyTorch framework is used. And to train the models’ data set prepared from different video clips with congested and uncongested traffic footage used. The research indicates that it is possible to detect traffic congestion from a surveillance camera in Ethiopia using one-stage object detectors.

Recommendations
• For this research, the data set is prepared from different uncontrolled video sources with different video quality mostly low-quality video which affects the accuracy of the model. But as feature work the model will achieve higher accuracy if video from a calibrated high-quality camera is used.
• For the future, the authors are interested to extend the work, so that the work can connect to the navigation system to provide information about the traffic congestion state using a mobile phone.
• As a recommendation, there are many papers written on traffic congestion detection using image processing techniques, but still, only a few researchers used the deep learning approach especially using one-stage object detectors. Currently, one-stage object detectors get the attention of many scholars because of their speed of processing with comparable accuracy with two-stage object detectors.

• As future work, the authors are interested to extend the work so that it can detect traffic congestion during night time and predicting traffic congestion before it happens.

Acknowledgments
Foremost, I would like to express my deepest and sincere gratitude to my advisor Dr. Kula Kekeba for the continuous support of my MSc thesis, for his patience, motivation, enthusiasm, and immense knowledge. His guidance helped me a lot in all parts of the thesis. Besides my advisor, I would like to thank every person who had contributed his part in any means to support me for the past two years. Finally, nothing is possible without the support and motivation of my bellowed family. Thank You All. . . .

Funding
Big data Journal provide publication fee waiver for African Countries

Availability of data and materials
Data set for this work available at http://dx.doi.org/10.17632/wtp4sswzd.1

Competing interests
The authors confirm that there are no competing interests. The article is not under any other review process and is not in the subject of any other submission.

Consent for publication
The authors consent for publication.

Authors’ contributions
B.B took the main role by conducting the literature review, implementation of the proposed model, conducting the experiments and writing the manuscript.
K.K took the supervisory role and oversaw the completion of the work and manuscript.

Author details
Department of Software Engineering, Big Data and HPC CoE, Addis Ababa Science and Technology University, Addis Ababa, Ethiopia.

References
1. G. Weisbrod, D. Vary, and G. Treyz, “Measuring economic costs of urban traffic congestion to business,” Transportation research record, vol. 1839, no. 1, pp. 98–106, 2003.
2. A. C. Sutandi, “Its impact on traffic congestion and environmental quality in large cities in developing countries,” in Proceedings of the Eastern Asia Society for Transportation Studies Vol. 7 (The 8th International Conference of Eastern Asia Society for Transportation Studies, 2009), pp. 180–180, Eastern Asia Society for Transportation Studies, 2009.
3. P. Bellini, S. Bilotta, M. Paolucci, M. Soderi, et al., “Real-time traffic estimation of unmonitored roads,” in 2018 IEEE 16th Intl Conf on Dependable, Autonomic and Secure Computing, 16th Intl Conf on Pervasive Intelligence and Computing, 4th Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech), pp. 935–942, IEEE, 2018.
4. S. Fadare and B. Ayantoyinbo, “A study of the effects of road traffic congestion on freight movement in lagos metropolis,” European Journal of Social Sciences, vol. 16, no. 3, pp. 429–437, 2010.
5. J. Nubert, N. G. Truong, A. Lim, H. I. Tanujaya, L. Lim, and M. A. Vu, “Traffic density estimation using a convolutional neural network,” arXiv preprint arXiv:1809.01564, 2018.
6. P. C. Commission et al., “Summary and statistical report of the 2007 population and housing census,” Population size by age and sex, 2008.
7. “Ministry of transport - ethiopia.”
8. G. F. Daie and L. Tebarek, “Identifying major urban road traffic accident black-spots (rtabs): A sub-city based analysis of evidences from the city of addis ababa, ethiopia,” Journal of Sustainable Development in Africa, vol. 15, no. 2, 2013.
9. H. Yilma, “Challenges and prospects of traffic management practices of addis ababa city administration,” MA Degree), Addis Ababa University School of Graduate Studies, Addis Ababa, 2014.
10. S. Puntavungkour and R. Shibasaki, “Vehicle detection from three line scanner image,” in Proceedings of the 2003 IEEE International Conference on Intelligent Transportation Systems, vol. 1, pp. 785–788, IEEE, 2003.
11. T. Chujoh and T. Watanabe, “Traffic density analysis apparatus based on encoded video,” June 1 2004. US Patent 6,744,908.
12. M. Liepins and A. Severdaks, “Vehicle detection using non-invasive magnetic wireless sensor network,” in 2013 21st Telecommunications Forum Telfor (TELFOR), pp. 601–604, IEEE, 2013.
13. M. Gholve and S. Chougule, “Traffic congestion detection for highways using wireless sensors,” *International Journal of Electronics and Communication Engineering*, vol. 6, pp. 259–265, 2013.

14. B. Barbagli, I. Magrini, G. Manes, A. Manes, G. Langer, and M. Bacchi, “A distributed sensor network for real-time acoustic traffic monitoring and early queue detection,” in *2010 Fourth International Conference on Sensor Technologies and Applications*, pp. 173–178, IEEE, 2010.

15. B. A. Alpatov, P. V. Babayan, and M. D. Ershov, “Vehicle detection and counting system for real-time traffic surveillance,” in *2018 7th Mediterranean Conference on Embedded Computing (MECO)*, pp. 1–4, IEEE, 2018.

16. P. Wang, L. Li, Y. Jin, and G. Wang, “Detection of unwanted traffic congestion based on existing surveillance system using in freeway via a cnn-architecture trafficnet,” in *2018 13th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, pp. 1134–1139, IEEE, 2018.

17. P. Soviany and R. T. Ionescu, “Optimizing the trade-off between single-stage and two-stage deep object detectors using image difficulty prediction,” in *2018 20th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC)*, pp. 209–214, IEEE, 2018.

18. S. Luo, C. Xu, and H. Li, “An application of object detection based on yolov3 in traffic,” in *Proceedings of the 2019 International Conference on Image, Video and Signal Processing*, pp. 68–72, 2019.

19. K. Zhao, X. Zhu, H. Jiang, C. Zhang, Z. Wang, and B. Fu, “Dynamic loss for one-stage object detectors in computer vision,” *Electronics Letters*, vol. 54, no. 25, pp. 1433–1434, 2018.

20. W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, “Ssd: Single shot multibox detector,” in *European conference on computer vision*, pp. 21–37, Springer, 2016.

21. W. Li and G. Liu, “A single-shot object detector with feature aggregation and enhancement,” in *2019 IEEE International Conference on Image Processing (ICIP)*, pp. 3910–3914, IEEE, 2019.

22. C.-T. Lam, H. Gao, and B. Ng, “A real-time traffic congestion detection system using on-line images,” in *2017 IEEE 17th International Conference on Communication Technology (ICCT)*, pp. 1548–1552, IEEE, 2017.

23. M. Sujatha, D. Renuga, and S. Bragadeesh, “Traffic congestion monitoring using image processing and intimation of waiting time,” *International Journal of Pure and Applied Mathematics*, vol. 115, no. 7, pp. 239–245, 2017.

24. M. S. Uddin, A. K. Das, and M. A. Taleb, “Real-time area based traffic density estimation by image processing for traffic signal control system: Bangladesh perspective,” in *2015 International Conference on Electrical Engineering and Information Communication Technology (ICEEICT)*, pp. 1–5, IEEE, 2015.

25. P. Deshmne, “Mrkfe: Designing cascaded map reduce algorithm for key frame extraction,” in *2017 International Conference on Information Communication and Embedded Systems (ICICES)*, pp. 1–6, IEEE, 2017.