Performance Comparison of Object Detection Models for Traffic Density Estimation Using CCTV Footages

Chelsie D. Bajamunde, Mark Harold B. Bien*, Thrisia A. Sarcilla, and Referendo D. Soriano
Electronics and Computer Engineering Department
Ateneo de Naga University
Naga City, 4400 Philippines
*roldbien@live.com

Date received: November 12, 2025
Revision accepted: March 16, 2026

Abstract

This study evaluated three object detection models for estimating traffic density on Elias Angeles St., Naga City, using mean average precision (mAP). The object detection model classified vehicles into five classes: private cars, jeepneys, trucks, motorcycles, and tricycles. Closed-circuit television (CCTV) footage was subjected to adaptive background subtraction and morphological opening to produce 320px × 320px images for use as a dataset for object detection models. Using Common Objects in Context means Average Precision at Intersection over Union (COCO mAP at IoU=50 as the metric for mAP. Faster Region-Convolutional Neural Network (Faster R-CNN) achieved the highest mAP of 92.54%, compared with You Only Look Once version 3 (YOLOv3) and Single Shot MultiBox Detector (SSD). In the traffic density estimation, vehicle size was accounted for; consequently, a private car was used as the standard vehicle type.

***Keywords**:* adaptive background subtraction, mean average precision, morphological opening, object detection model, traffic density

1. Introduction

Transportation plays a significant role in human civilization and urbanization. As an instrument for transporting people and goods from one location to another, it shapes the economic, social, political, and cultural dimensions of the world. In fact, urbanization and urban development occur in line with the development of urban transport systems, as is evident in Naga City. Since an urban area is composed of a city and the surrounding municipalities, transportation must be able to keep up with the growing demands of urbanization. Inability to meet the rising demand for effective and efficient transportation, especially in urban areas such as cities, would significantly

affect productivity and even contribute to the existing problems, such as traffic congestion and parking.

As noted by Chin *et al*. (2017), traffic congestion occurs when road capacity is exceeded, resulting in slower traffic flow and longer travel times for commuters. In addition, poorly planned road networks, rapid population growth, and inadequate traffic management were considered factors contributing to traffic congestion.

Therefore, effective and efficient traffic management is needed to ensure predictable and reliable transportation for people and goods. These traffic management services require data to generate short-term traffic flow forecasts and other data that road users need, such as traffic density, required vehicle speeds, and alternative routes, which are used in response to unexpected road incidents and events. The alternative routes are electronically controlled.

Various studies were conducted to identify the specific factors affecting traffic and to determine the other causes of traffic jams or congestion. One of the research projects, such as Moridpour *et al*. (2014), focused on the effects of heavy vehicles on traffic flow. With this, the researchers sought to examine the effects of different vehicles on traffic density.

To further explore solutions for traffic management and traffic-jam mitigation, the researchers aim to develop a system that estimates traffic density using vehicle-type identification and vehicle counts. Specifically, the study aimed to integrate video-processing techniques and use existing CCTV footage to assess traffic density. Such techniques were used to design an algorithm that provides dataset for the three object detection models gathered from the CCTV footages: You Only Look Once version 3 (Redmon & Farhadi, 2018), Single-Shot MultiBox Detector (Liu *et al.,* 2016), and Faster Region-Convolutional Neural Network (Ren *et al*., 2017). Object detection is a vision technique that identifies objects in an image, classifies the objects, and locates them in the image. In addition, Arlen (2018) emphasized that object detection is a process used by computer systems that combines localization and classification tasks.

Afterwards, the three object detection models were compared in terms of their accuracy to determine the suitable object detector in the system. Finally, the vehicle count from the object detector was utilized for traffic density estimation.

2. Methodology

To estimate the traffic density in Naga City, the flow is composed of nine modules, namely: CCTV Footages Acquisition, Creation of Dataset, PSO Coordination, Image Labeling, Training and Evaluation of Object Detection Models, Comparison of Object Detection Models Performance Parameters, Vehicle Count Computation, Generation of Hourly Traffic Flow Estimation Report, and System Integration.

For the start of project implementation, the researchers requested fifteen days of traffic CCTV footage from the Central Communication Center (COMCEN) of the Naga City local government. However, the CCTV was 40 meters from the target street, operated at 20 frames per second (fps), and lacked a night mode. Therefore, this study covers only periods during which the vehicles are visually identifiable in the CCTV footage, from 5:00 AM to 6:00 PM.

The task of recognizing moving vehicles in the videos played a vital role in the dataset creation for training and evaluating object detection models to be compared in the study. In this context, adaptive background subtraction and morphological opening were employed to detect moving vehicles in the video and reduce noise. Moreover, this process saves images of identified moving vehicles at 320×320 pixels for dataset creation and ensures that as many vehicles as possible can be accommodated within this dimension, thereby reducing the likelihood of duplicate images.

The researchers coordinated with the Public Safety Office – Traffic Management Committee regarding the standard process and techniques used by the office to conduct traffic flow analysis. As a result, the vehicles in the images in the created dataset were labeled in accordance with the recognized standard vehicle types in Naga City, which are Car, Truck, Motorcycle, Trimobile, and Public Utility Jeepney. Labels assigned to the vehicles present in the images is a pre-processing procedure done to prepare the dataset to be used in the training and evaluation of the chosen object detection models.

For the vehicle classification aspect of the system, three object detection models were compared in response to the target of finding the suitable object detector for the desired system. These object detectors were trained using the created dataset and evaluated according to its speed, Mean Average Precision (mAP) and Intersection over Union (IoU) parameters. Once the following parameters of the three object detectors were compared and the suitable model

determined, the chosen object detector was used to compute the vehicle count per vehicle type.

The vehicle count computed per vehicle type functioned as a crucial step for achieving the main goal of the system which is to generate an hourly traffic flow estimation report. Once the system can generate its required output, system integration would be emulated through HTTP Streaming. The entire flow can be seen in Figure 1.

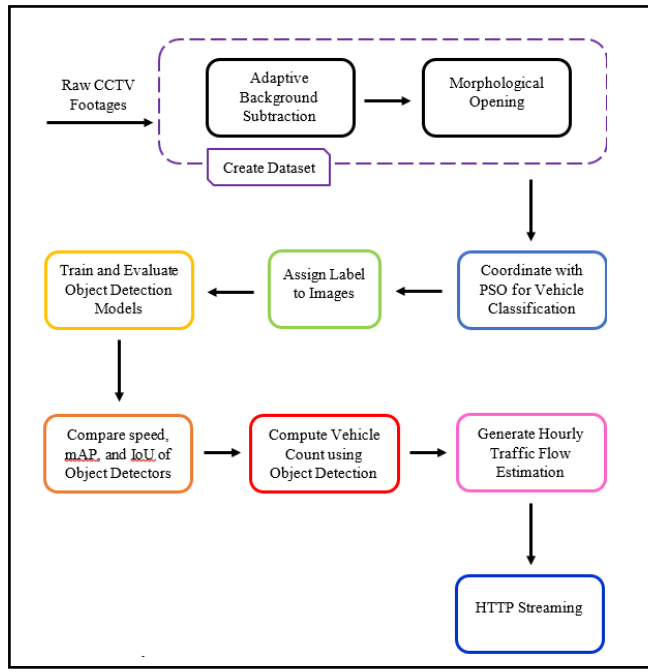

Figure 1. Project process flowchart

2.1 Dataset Gathering

The study focused on estimating traffic density on Elias Angeles Street in Naga City. The study also focused only on the following vehicles: motorcycles, tricycles, jeepneys, cars, and trucks.

The twelve days of raw traffic CCTV footage used in this study were requested from the Central Communication Center (COMCEN) of the Naga City local government. Videos from November 22-30, 2018, and December 13, 2018, were used to generate the dataset, while videos from December 11-12, 2018,

were used to evaluate the overall system efficiency. The days used for dataset generation and system evaluation were selected at random to ensure the accuracy of the entire system.

The CCTV footage used was limited to 5:00 AM to 5:00 PM, as it is challenging to identify vehicle types outside those hours, even to the naked eye, due to the glare from vehicle headlights. The CCTV footage is recorded at 20 frames per second.

2.2 Video Processing

The system employed object detection for vehicle classification. In return, training and evaluation datasets for object detection were generated from CCTV footage via video processing. The primary goal of preprocessing the videos was to maximize the number of saved images while reducing noise and false positives. In this case, the vehicles in the video were extracted from the frame and saved as a 320px × 320px image, which was the recommended image size for the images fed into the neural networks used in the study, as shown in Figure 2. The video processing techniques were entirely done using MATLAB (The MathWorks Inc., n.d.) for dataset generation and a custom program in Visual C# (Microsoft, n.d.) for dataset labeling.

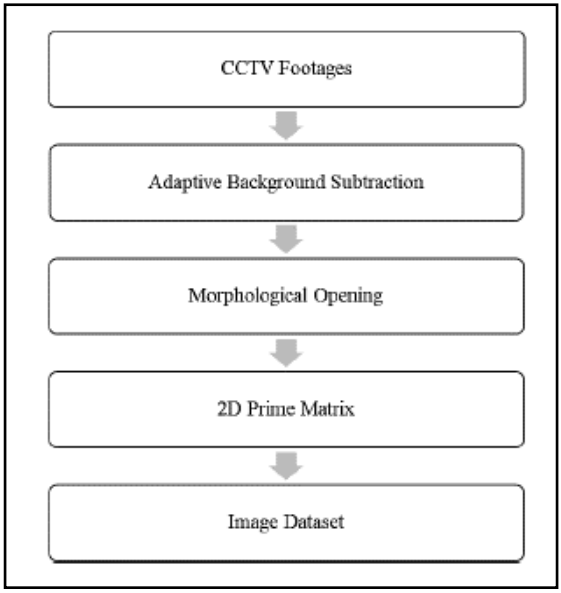

Figure 2. Video processing for object detection dataset

2.2.1 Adaptive Background Generation

To extract images from the CCTV footage for the image dataset, the background was generated from the video to extract the foreground (vehicles). This background has two properties to create a good image dataset: accuracy of the initial background and resilience to changes in the environment, especially during sunrise and sunset, wherein there is a gradual change in the environment. To create an initial background, the algorithm focused on the accuracy of the background rather than limiting itself to a set of frames for it to be compared. The algorithm for creating the initial background works by initially subtracting the first two frames of the video, then multiplying it to the threshold values of hue and value, creating a two-dimensional matrix of ones and zeroes wherein one is the accepted pixel and 0 is the failed pixel or a potential object which is not needed in creating an initial background. This is then multiplied to the current frame and will result in a three-dimensional matrix of accepted pixels from the current frame.

The algorithm keeps on finding new accepted pixels by having another frame to be compared to each of the used frames until it reaches a certain value which is called Minimum Averaged Frame per Pixel (MAFP) since two frames will not be enough in creating a full background as shown in Figure 3.

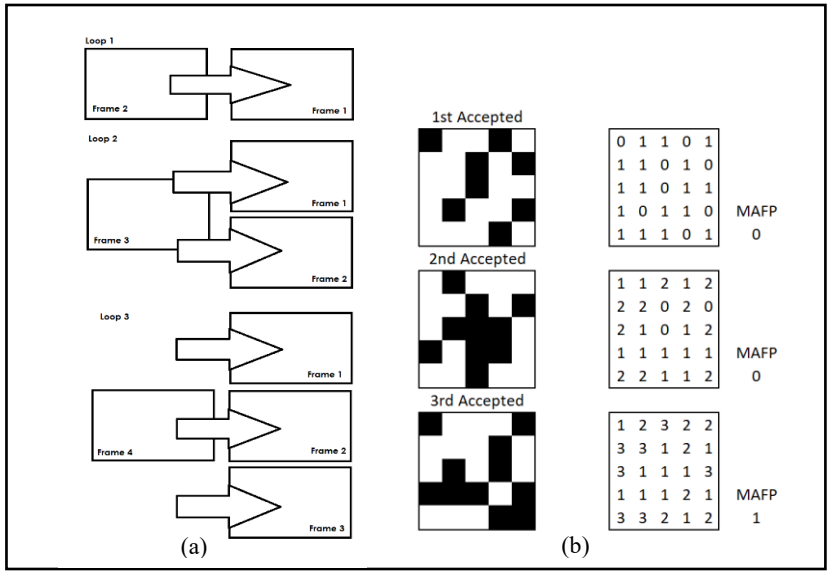

Figure 3. Looping for finding new accepted pixels (a) and Minimum Averaged Frame per Pixel (MAFP) (b)

After reaching the recommended minimum averaged frame per pixel, the acquired frames undergo averaging pixel per pixel depending on the value of each pixel which is based on its own magnitude in the placeholder of the MAFP in order to create an initial background with a good confidence that each pixel has at least appeared in n number of frames and could be confirmed that the pixel in the initial background is not an object. An initial background generated using such an algorithm is seen in Figure 4.

Figure 4. Initial background generated from the study route

After the initial background is created, this background should constantly adapt to gradual changes in the video. This includes sunrise and sunset hours, and the weather changes from sunny to cloudy. It was achieved by updating the background to the minor changes in the current frame. Each background pixel in the current frame will have a specific effect on the background to have adaptability properties. This background update can be seen in Figure 5, herein the road became slightly brighter due to the change to a brighter sky. Other parts of the background, such as the light from streetlamps, did not update due to the abrupt change, even though the lights are supposedly closed at the new background. Nonetheless, it did not affect the quality of the image dataset as these streetlamps are far from the road that the researchers are interested in.

(a) (b)

Figure 5. Transition from old background (a) to new background (b)

2.2.2 Adaptive Background Subtraction

After the adaptive background is generated, current frames are subtracted from the background and converted into a 2D matrix of black (0) or white (1) pixels based on a threshold value, as shown in Figure 6. By doing this, objects that are not in the background will show up in the foreground. Since the background that was created excludes any potential object, by subtracting the current frame from the background, vehicles moving along the road will show up in the foreground.

Figure 6. Frame subtracted to adaptive background

2.2.3 Region of Interest and Morphological Opening

Most background noise was removed by the background processing. However, some unnecessary blobs and salt noise from the CCTV footage need further processing, as seen in Figure 6. To further minimize the noise from the frame, two techniques were applied: Region of Interest (ROI) and Morphological Opening.

Although CCTV was directly looking at the street of interest, only a certain region of the video (main road) is needed. By removing the non-essential regions in the frame, it would increase the speed of the system and at the same time remove most of the excess noise in the frame coming from the environment. In the videos, only the visible parts of the study were included in the region of interest which is shown in Figure 7. The ROI is only a 2D matrix of zeros and ones pertaining to the non-essential and essential regions respectively but was visualized for reference. This ROI is multiplied to the subtracted frame to remove all the noise outside the ROI.

Figure 7. Region of interest

Once the ROI is applied to the current frame, only noise from the street itself is left, as seen in Figure 8 (left). The nature of the street noise is small blobs, so a simple morphological opening was applied to remove them, leaving images with minimal to no noise, as shown in Figure 8 (right). This was achieved by using *bwareaopen()* in the MATLAB Image Processing and Computer Vision Toolbox.

Figure 8. Before (a) and after (b) applying morphological opening

2.2.4 2-Dimensional Prime Matrix

After removing majority of the noise in the frame, each object could now be placed in a 320px × 320px frame for labeling. However, the system can still be further optimized by determining whether multiple objects can fit inside the 320px × 320px frame. Such an issue can be seen in Figure 9 wherein red and orange boxes are simulated as two different nearby objects.

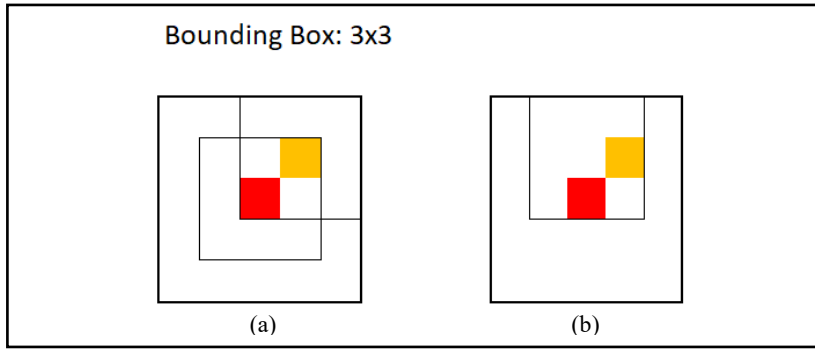

Figure 9. Redundancy of boxing objects (a) and the proposed algorithm (b)

This algorithm can be implemented by constructing a 2D prime matrix. Each object is assigned a unique prime number and then multiplied by an array of ones depending on the location of the object, as if it were bounded by a 320px × 320px frame alone. After all the prime numbers of each object are multiplied by the 2d prime matrix, each unique number in the matrix will be factored starting from the most significant number. Since all objects are prime numbers, each factor of the unique number is the representation of each object, and it is checked whether all objects will fit in the 320px × 320px frame. If all objects fit within the frame, they will be centered and included in the image

dataset. This is done until the lowest number in the matrix. The algorithm is visualized in Figure 10.

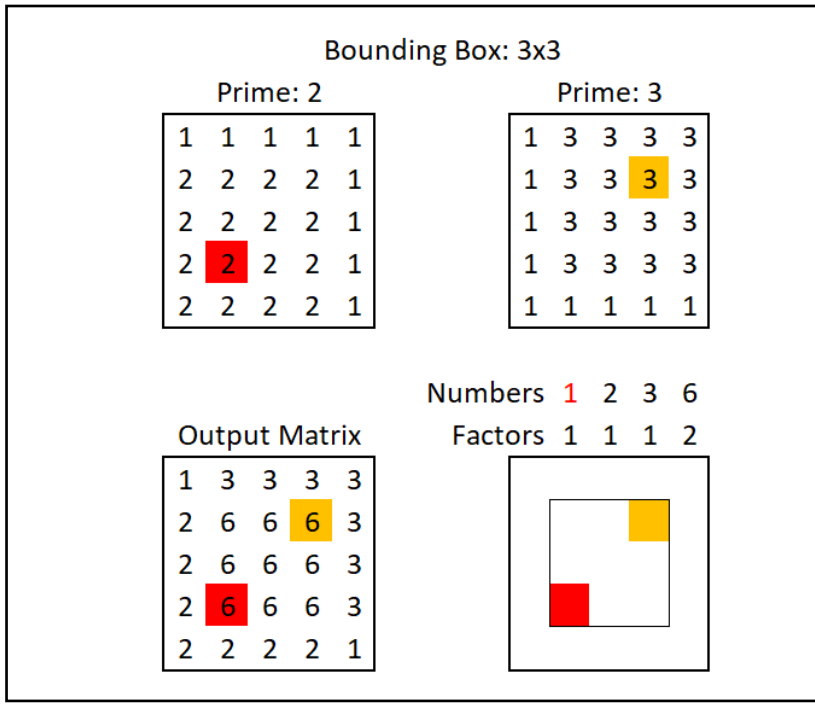

Figure 10. 2D Prime matrix and boxing method of foreground objects

2.2.5 Image Labeling

After all the pre-processing for the images were done, a total of 8225 images were gathered and labeled. The images in the dataset need to be labeled before feeding it to the object detection models. The labeled data would serve as the guide for the object detection models of what to learn and to provide in the outside layer of the models.

Each vehicle is labeled using programs developed by the researchers tailored for the study. Two programs were developed for the study: An image labeler for the five vehicle types and a label converter between YOLO v3 labels and PASCAL VOC XML used by Faster R-CNN and SSD. An example of labeled data with its corresponding text file is shown in Figure 11.

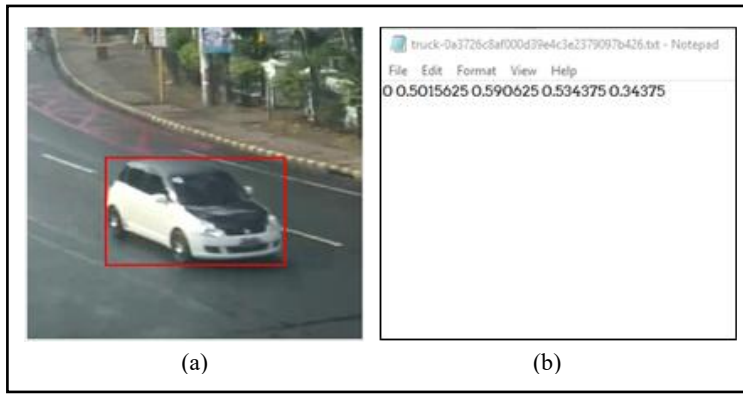

Figure 11. Labeled data and corresponding text file

Because of the 2D prime matrix, multiple vehicles can exist in a single picture. From the 8225 images, a total of 22219 labels were generated. Table 1 shows the labels generated from the image dataset.

Table 1. Labels generated from image dataset

Class	Total Label
Truck	412
Private Car	5032
Motorcycle	5323
Tricycle	10859
Public Utility Jeepney (PUJ)	593
TOTAL	22219

2.3 Object Detection

In the desire to utilize a suitable object detector for the task of vehicle classification, the researchers compared three object detectors: YOLO v3, SSD, and Faster R-CNN.

YOLO v3 incorporates residual blocks, skips connections, and upsampling, though at a slight cost to speed compared to its predecessor as seen in Figure 12. It uses Darknet-53 as its feature extractor and makes predictions at three different scales (32, 16, and 8) with three bounding box anchors per scale, totaling nine anchors. It employs binary cross-entropy loss for classification and uses independent logistic classifiers for multi-label classification, allowing multiple non-exclusive labels per object if their scores exceed a set threshold (Redmon & Farhadi, 2018).

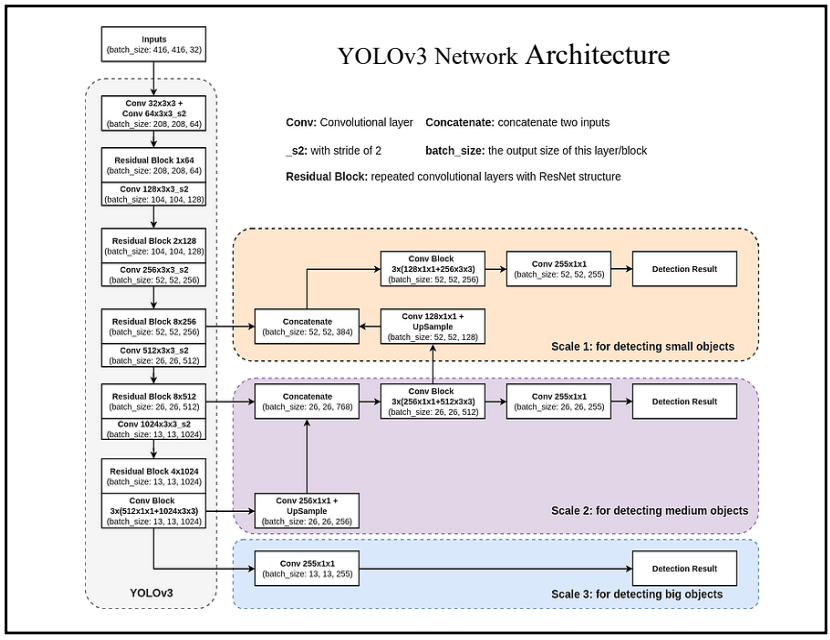

Figure 12. YOLOv3 network architecture (Prusty *et al*., 2021)

SSD (Single Shot MultiBox Detector) is a fast and efficient feed-forward convolutional network for object detection, based on the VGG-16 architecture for strong feature extraction and transfer learning. It incorporates auxiliary convolutional layers to detect objects at multiple scales while progressively reducing input dimensions as seen in Figure 13. SSD achieved a strong balance between speed and accuracy. Unlike traditional methods, it processes an image in a single pass, using anchor boxes of varying aspect ratios and predicting bounding box offsets rather than direct coordinates. This design makes it one of the best real-time object detectors in terms of speed-accuracy trade-offs (Liu *et al*., 2016).

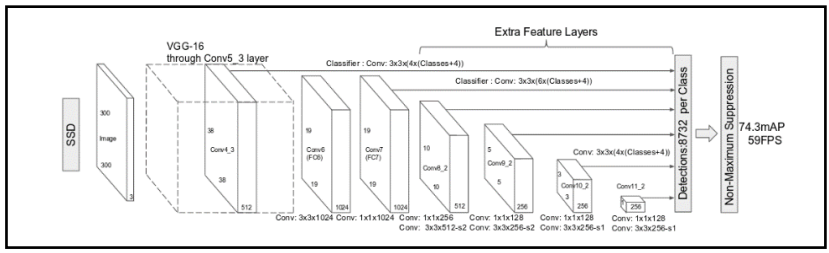

Figure 13. Single Shot MultiBox Detector Network Architecture (Liu *et al*., 2016)

Faster R-CNN is an advanced object detection model that evolved from R-CNN and Fast R-CNN, improving speed and efficiency. Faster R-CNN integrates a Region Proposal Network (RPN) to generate region proposals internally, eliminating the need for external methods. The RPN uses anchor boxes at three scales and aspect ratios (totaling nine per location) to classify regions as foreground or background as seen in Figure 14. By sharing convolutional features between the RPN and detection network, Faster R-CNN achieves faster, and more accurate object detection compared to its predecessors (Ren *et al*., 2017).

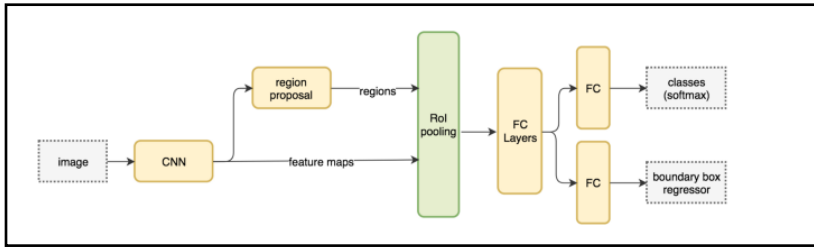

Figure 14. Faster R-CNN Network Architecture (Hui, 2018)

The object detectors are evaluated for their accuracy. Once the whole images in the dataset were hand-labeled, the dataset was split into the Training Dataset (80%) and the Evaluation Dataset (20%). This split is based upon the data to be used for the models to avoid overfitting and underfitting (Sivakumar *et al*., 2024)

Transfer learning was utilized for each object detector for better and faster training. The initial weights used for the training were the pre-trained weights gathered when the models were pre-trained with COCO 2017, a large-scale object detection dataset with 80 object categories. COCO 2017 and the dataset generated by this study had similar classes – private car, truck, and motorcycle.

The object detectors had trained for different lengths of time since each model uses a different loss function - a part of a neural network that measures error in predictions. Moreover, each object detection model uses a different feature extractor; YOLO v3 employs its own, Darknet-53. SSD used Inception V2, the second refinement of the Inception architecture. Faster R-CNN used ResNet-101, a convolutional network trained on the ImageNet dataset. A general comparison of the three object detection models is presented in Table 2.

Table 2. Object detection models training details

Training Details	YOLO v3	SSD	Faster R-CNN
Feature Extractor	Darknet-53	Inception V2	Resnet 101
Initial Pre-trained Weights	COCO 2017	COCO 2017	COCO 2017
Training Hours	20	14	9

Bounding boxes were used to visualize object detection in the five classes. In Figure 15, four vehicles were detected by the object detector. Specifically, two tricycles (white box), a car (green box), and a PUJ (light-blue box) were detected within the region of interest. The object detection model did not account for other vehicles outside the region of interest, such as those located beyond the pedestrian lane.

Figure 15. Vehicle detection with bounding boxes

2.3 Traffic Density Estimation

Traffic density dramatically varies on the vehicles currently on the roadway. Numerous smaller vehicles may have the same effect on traffic as a smaller number of larger vehicles. This makes the traffic density highly variable to the size of the vehicles. With this, Equation 1 is used to calculate traffic density considering the sizes of different vehicles.

$$TD = \frac{v_{private}\beta_{private}+v_{puj}\beta_{puj}+v_{truck}\beta_{truck}+v_{motor}\beta_{motor}+v_{trike}\beta_{trike}}{d} \quad (1)$$

where *TD* is the traffic density*;* $v_{private}$*,* v_{puj}*,* v_{truck}*,* v_{motor}*,* and v_{trike} is the number of private cars, public utility jeepneys, trucks, motorcycles, and tricycles

respectively; $\beta_{private}$, β_{puj}, β_{truck}, β_{motor}, and β_{trike} is the private car, public utility jeepney, trucks, motorcycle, and tricycle multipliers respectively.

Multipliers of each vehicle type were based upon the road area covered by the vehicle. The standard used in the multipliers was the private car, and all other multipliers were based on the ratio of the road area covered by other vehicles to the private car. Table 3 shows the vehicle count multipliers used in Equation 1.

Table 3. Vehicle count multipliers

Vehicle Type	β
Private Car	1
Public Utility Jeepney (PUJ)	2.467319
Truck	2.803772
Motorcycle	0.137372
Tricycle	0.191018

2.4 HTTP Streaming

To emulate the setup from COMCEN, this study focused on integrating the system without any changes in their current system. This was achieved through utilizing HTTP Streaming. Figure 16 illustrates the setup of emulating the integration of the developed system. The server computer and the computer with the system were connected to the router. Once connected to the Local Area Network, the server computer would be sending a stream of images as a motion jpeg (m-jpeg) file. The streamed m-jpeg file will be used as the input for the object detector in the computer with the traffic density estimation system.

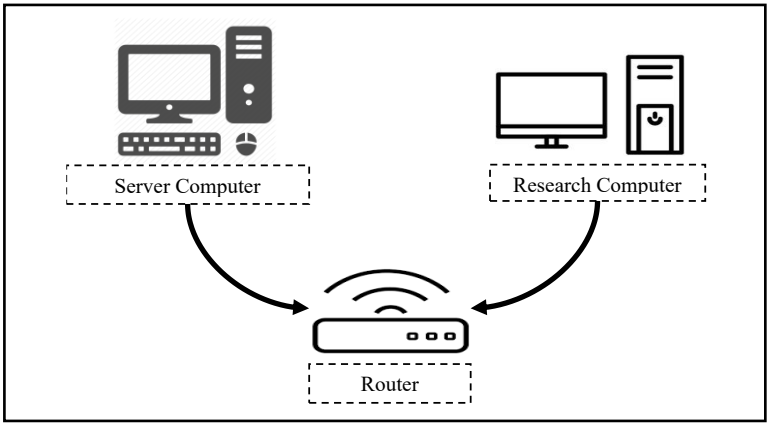

Figure 16. Physical setup

3. Results and Discussion

To create the ideal system for traffic density estimation, multiple evaluations were conducted, starting from pre-processing of video footage, selecting the ideal object detection model, and comparing the model to actual manual count, which is commonly done by the Public Safety Office.

3.1 Comparison Between Adaptive Background Subtraction and Background Subtraction

A difference between normal background subtraction and adaptive background subtraction is about 0.022% in normal lighting conditions, with low or almost no change of environment. It is observed that the difference between the two methods increased as time progressed, since the normal background subtraction did not update itself, but the environment changed as time progressed. Moreover, the adaptive background subtraction stays low even if the normal background subtraction increases, as shown in Figure 17. Yellow and blue lines show the percentage difference in full video frame between current frame and background in when normal and adaptive background subtraction was used, respectively. On the other hand, purple and red lines show the percentage difference outside the region of interest when normal and adaptive background subtraction was used respectively to highlight the transitional changes of the background itself.

Another observation is that full-frame samples produced more spikes than those without ROI, due to vehicles on the road. Both adaptive background and normal background considered the presence of vehicles as a factor affecting the percentage differences of frames, resulting in an abrupt increase when there is a vehicle, especially in a traffic jam.

3.2 Performance Comparison of Object Detection Models

Mean Average Precision (mAP) is an evaluation metric used to assess the accuracy of object detectors. Mean Average Precision was calculated with the average precision values per class from the three object detection models, along with their details as shown in Table 4. Mean Average Precision can also be evaluated for each class individually. Faster R-CNN, a region-based object detector, achieved the highest accuracy among the three object detectors, with a mean average precision of 92.54%.

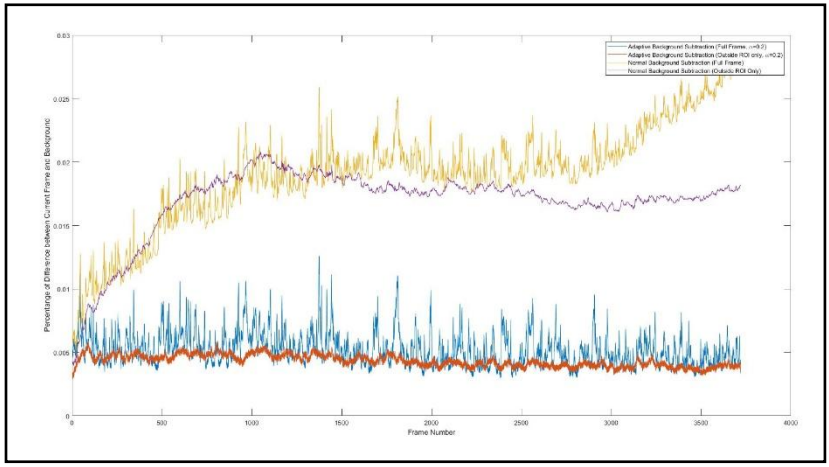

Figure 17. Comparison between Normal Background Subtraction and Adaptive Background Subtraction with low change of environment

Table 4. Summary of Mean Average Precision of different object detection models

Training Details	Object Detection Models		
	YOLO v3	SSD	Faster RCNN
Feature Extractor	Darknet-53	Inception V2	Resnet 101
Initial Pre-trained Weights	COCO 2017	COCO 2017	COCO 2017
Training Hours	20	14	9
Class	Object Detection Models		
	YOLO v3 (mAP)	SSD (mAP)	Faster RCNN (mAP)
Car	89.78	84.13	94.93
PUJ	87.64	78.05	91.95
Truck	82.94	81.79	86.63
Motorcycle	88.74	79.87	94.01
Tricycle	90.36	82.2	95.19
COCO mAP @IoU = 50	87.89	81.21	92.54

3.3 Percentage Error

A comparison of the object detector's performance with the actual vehicle count on the road was conducted by calculating the percentage error between the theoretical and actual traffic density values, as shown in Table 5. The manually calculated traffic densities were treated as theoretical values. In contrast, the traffic densities calculated by the system were assigned as the actual values for calculating percentage error. Both methods used Equation 1 to calculate traffic density considering the sizes of different vehicles present in the study routes.

Table 5. Percentage error between theoretical and experimental values

Date	Early Morning (%)	Late Morning (%)	Early Afternoon (%)	Late Afternoon (%)
Nov 22	10.204	2.131	2.131	(No footage)
Nov 23	6.030	5.162	1.153	3.981
Nov 24	11.558	13.586	5.081	9.900
Nov 25	12.053	12.820	7.639	0.045
Nov 26	21.454	24.322	17.334	21.814
Nov 27	24.608	18.822	4.158	21.301
Nov 28	1.953	10.800	17.649	15.691
Nov 29	2.786	11.662	4.075	2.406
Nov 30	3.477	14.229	1.186	6.758
Dec 13	10.207	4.463	3.946	10.138

The study considered four (4) time period divisions: early morning (5:00 AM – 8:00 AM), late morning (9:00 AM – 12:00 NN), early afternoon (1:00 PM – 2:30 PM), and late afternoon (4.30 PM – 5:00 PM), with peak hours at early morning and late afternoon since the road belongs to a school zone.

4. Conclusion and Recommendation

Overall system performance has an average percentage error of 10.03%, which is acceptable given that the selected object detector, Faster R-CNN, achieves a mean average precision of 92.54%.

Factors affecting the percentage error included the data trained in the neural network, which focused on videos with less traffic. As a result, vehicles hidden behind other vehicles would not be recognized by the models. Another observation gathered from the research was that the object detector model tends to have difficulties with low-light and blurry videos and false detections outside the road mainly due to the limitations of the CCTV used in the research. An increase in the number of image representations per class in the dataset and application of region of interest until the object detection model reaches stability in detecting and classifying the vehicles contributes to increasing the accuracy and efficiency of the overall system. To further improve the accuracy of the model, image samples should be gathered on all streets with CCTV cameras installed, along with a more robust object tracking algorithm to track vehicles even behind other vehicles.

5. Acknowledgement

The authors would like to thank the local government unit of Naga City, Philippines, specifically the Central Communication Center for allowing them to use the CCTV footages in the study.

6. References

Arlen, T. (2018). Understanding the mAP Evaluation Metric for Object Detection. Retrieved from https://medium.com/@timothycarlen/understanding-the-map-evaluation-metric-for-object-detection-a07fe6962cf3

Chin, V., Jaafar, M., Moy, J., Phong, M., Wang, S., McDonnell, M., & Prawiradinata, I. (2017). Unlocking Cities: The impact of ridesharing in Southeast Asia and beyond. *The Boston Consulting Group*.

Hui, J. (2018). What do we learn from region based object detectors (Faster R-CNN, R-FCN, FPN)*?*. Retrieved from https://jonathan-hui.medium.com/what-do-we-learn-from-region-based-object-detectors-faster-r-cnn-r-fcn-fpn-7e354377a7c9

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A.C. (2016). SSD: Single shot multibox detector. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 9905 LNCS*. https://doi.org/10.1007/978-3-319-46448-0_2

MathWorks Inc. (2018). MATLAB release notes for R2018a [Software]. Retrieved from https://www.mathworks.com/help/release-notes/r2018a.html

Microsoft (n.d.). Visual Studio [Software]. Retrieved from https://visualstudio.microsoft.com/

Moridpour, S., Mazloumi, E., & Mesbah, M. (2015). Impact of heavy vehicles on surrounding traffic characteristics. *Journal of Advanced Transportation*, *49*(4), 535–552. https://doi.org/10.1002/atr.1286

Prusty, M., Tripathi, V., & Dubey, A. (2021). A novel data augmentation approach for mask detection using deep transfer learning. *Intelligence-Based Medicine*, *5*, 100037. https://doi.org/10.1016/j.ibmed.2021.100037

Redmon, J., & Farhadi, A. (2018). YOLOv3: An Incremental Improvement. *arXiv*. https://doi.org/10.48550/arXiv.1804.02767

Ren, S., He, K., Girshick, R., & Sun, J. (2017). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *39*(6). https://doi.org/10.1109/TPAMI.2016.2577031

Sivakumar, M., Parthasarathy, S., & Padmapriya, T. (2024). Trade-off between training and testing ratio in machine learning for medical image processing. *PeerJ Computer Science*, *10*, e2245. https://doi.org/10.7717/peerj-cs.2245