Hindawi
Mathematical Problems in Engineering
Volume 2021, Article ID 1577614, 10 pages
https://doi.org/10.1155/2021/1577614

Research Article

A Real-Time Vehicle Counting, Speed Estimation, and Classification System Based on Virtual Detection Zone and YOLO

Cheng-Jian Lin,[1,2] **Shiou-Yun Jeng**,[3] **and Hong-Wei Lioa**[1]

[1]*Department of Computer Science and Information Engineering, National Chin-Yi University of Technology, Taichung 411, Taiwan*
[2]*College of Intelligence, National Taichung University of Science and Technology, Taichung 404, Taiwan*
[3]*Department of Business Administration, Asia University, Taichung 413, Taiwan*

Correspondence should be addressed to Cheng-Jian Lin; cjlin@ncut.edu.tw

Received 7 May 2021; Revised 30 July 2021; Accepted 7 October 2021; Published 2 November 2021

Academic Editor: Teen-Hang Meen

In recent years, vehicle detection and classification have become essential tasks of intelligent transportation systems, and real-time, accurate vehicle detection from image and video data for traffic monitoring remains challenging. The most noteworthy challenges are real-time system operation to accurately locate and classify vehicles in traffic flows and working around total occlusions that hinder vehicle tracking. For real-time traffic monitoring, we present a traffic monitoring approach that overcomes the abovementioned challenges by employing convolutional neural networks that utilize You Only Look Once (YOLO). A real-time traffic monitoring system has been developed, and it has attracted significant attention from traffic management departments. Digitally processing and analyzing these videos in real time is crucial for extracting reliable data on traffic flow. Therefore, this study presents a real-time traffic monitoring system based on a virtual detection zone, Gaussian mixture model (GMM), and YOLO to increase the vehicle counting and classification efficiency. GMM and a virtual detection zone are used for vehicle counting, and YOLO is used to classify vehicles. Moreover, the distance and time traveled by a vehicle are used to estimate the speed of the vehicle. In this study, the Montevideo Audio and Video Dataset (MAVD), the GARM Road-Traffic Monitoring data set (GRAM-RTM), and our collection data sets are used to verify the proposed method. Experimental results indicate that the proposed method with YOLOv4 achieved the highest classification accuracy of 98.91% and 99.5% in MAVD and GRAM-RTM data sets, respectively. Moreover, the proposed method with YOLOv4 also achieves the highest classification accuracy of 99.1%, 98.6%, and 98% in daytime, night time, and rainy day, respectively. In addition, the average absolute percentage error of vehicle speed estimation with the proposed method is about 7.6%.

1. Introduction

Traffic monitoring with an intelligent transportation system provides solutions to various challenges, such as vehicle counting, speed estimation, accident detection, and assisted traffic surveillance [1–5]. A traffic monitoring system essentially serves as a framework to detect the vehicles that appear on a video image and estimate their position while they remain in the scene. In the case of complex scenes with various vehicle models and high vehicle density, accurately locating and classifying vehicles in traffic flows is difficult [6, 7]. Moreover, limitations occur in vehicle detection due to environmental changes, different vehicle features, and relatively low detection speeds [8]. Therefore, an algorithm must be developed for a real-time traffic monitoring system with the capabilities of real-time computation and accurate vehicle detection. Therefore, the accurate and quick detection of vehicles from traffic images or videos has theoretical and practical significance.

With the rapid development of computer vision and artificial intelligence technologies, object detection algorithms based on deep learning have been widely investigated. Such algorithms can extract features automatically through machine learning; thus, they possess a powerful image abstraction ability and an automatic high-level feature representation capability. A few excellent object detection networks, such as single-shot detection (SSD) [9], Fast R-CNN [10], YOLOv3 [11], and YOLOv4 [12], have been implemented for traffic detection using deep learning object detectors [13]. For example, Biswas et al. [14] implemented SSD to estimate traffic density. Yang et al. [15] proposed a multitasking-capable Faster R-CNN method that uses a single image to generate three-dimensional (3D) space coordinate information for an object with monocular vision to facilitate autonomous driving. Huang et al. [8] proposed a single-stage deep neural network called YOLOv3 and applied it to data sets generated in different environments to improve its real-time detection accuracy. Hu et al. [16] proposed an improved YOLOv4-based video stream vehicle target detection algorithm to solve the problem in the detection speed. In addition, the most noteworthy challenges associated with traffic monitoring systems are real-time operation for accurately locating and classifying vehicles in traffic flows and total occlusions that hinder vehicle tracking. Therefore, YOLO was developed as a regression-based, high-performance algorithm for the real-time detection of and statistics collection from vehicle flows.

The robustness of YOLOv3 and YOLOv4 to road marking detection improves its accuracy in small target detection. The model that is based on the TensorFlow framework, to enhance the real-time monitoring of traffic-flow problems by an intelligent transportation system [17]. The YOLOv3 network comprises 53 layers. It uses the Feature Pyramid Network for pedestrian detection to handle general multiscale object detection problems and the deep residual network (ResNet) ideas to extract image features for achieving a trade-off between detection speed and detection accuracy [18]. In addition to leveraging anchor boxes with predesigned scales and aspect ratios to predict vehicles of different sizes, YOLOv3 and YOLOv4 can realize real-time vehicle detection with a top-down architecture [19]. Moreover, a real-time vehicle detection and classification system can perform foreground extraction, vehicle detection, vehicle feature extraction, and vehicle classification [20]. To test the proposed method for vehicle classification, a vehicle-feature-based virtual detection zone and virtual detection line, which are predefined for each frame in a video, are used for vehicle feature computation [21]. Grents et al. [22] proposed a video-based system that uses a convolutional neural network to count vehicles, classify vehicles, and determine the vehicle speed. Tabassum et al. [23, 24] applied YOLO and a transfer learning approach to recognize native vehicles and vehicle classification on Bangladeshi Roads. Therefore, YOLO can be used to obtain a better matching map.

To increase vehicle counting and classification problems in real-time traffic monitoring, this study presents a real-time traffic monitoring system based on a virtual detection zone, Gaussian mixture model (GMM), and YOLO to increase the vehicle counting and classification efficiency. GMM and a virtual detection zone are used for vehicle counting, and YOLO is used to classify vehicles. Moreover, the distance and time traveled by a vehicle are used to estimate the speed of the vehicle. The major contributions of this study are described as follows: (1) A real-time traffic monitoring system is developed to perform real-time vehicle counting, vehicle speed estimation, and vehicle classification; (2) the virtual detection zone, GMM, and YOLO are used to increase vehicle counting and classification efficiency; (3) the distance and time traveled by a vehicle is proposed to estimate the vehicle speed; and (4) the MAVD, GRAM-RTM, and our collection data sets are used to verify various methods and the proposed method with YOLOv4 achieving the highest classification accuracy in the three data sets.

The remainder of this study is organized as follows. Section 2 describes the materials and methods, including data set preparation, vehicle counting method, and vehicle classification method. Section 3 presents the results of and a discussion on the proposed real-time vehicle counting, speed estimation, and classification system based on a virtual detection zone and YOLO. Finally, Section 4 presents a few concluding remarks and an outline for future research on real-time traffic monitoring.

2. Materials and Methods

To count vehicles from traffic videos, this study proposes a real-time vehicle counting, speed estimation, and classification system based on the virtual detection zone and YOLO. We combined a vehicle detection method with a classification system on the basis of two conditions between the virtual detection zone and the virtual detection lane line. To detect vehicles, a Gaussian mixture model (GMM) is applied to detect moving objects in each frame of a traffic video. Figure 1 shows a flowchart of the vehicle counting and classification process used in the proposed real-time vehicle counting, speed estimation, and classification system. In this study, first, traffic videos are collected to train the image data and used to perform vehicle classification verification. Next, GMM and virtual detection zone are used for vehicle counting. Finally, YOLO is used to perform vehicle classification in real time. In this study, the three steps are described as follows:

Part 1: Collect traffic videos from online cameras.

In this study, traffic videos were collected from online cameras and used for image data training and vehicle classification verification, as described in Section 2.1.

Part 2: Perform vehicle counting using GMM and virtual detection zone.

To realize real-time vehicle counting, object detection and recognition are performed. A virtual detection lane line and virtual detection zone are used to perform vehicle counting and speed estimation, respectively, as described in Section 2.2 and Section 2.4, respectively.

Part 3: Perform vehicle classification and speed estimation using the YOLOv3 and YOLOv4 algorithms.

2.1. Data Set Preparation

2.1. Data Set Preparation. The data set used in this study was prepared by collecting traffic videos recorded with online cameras installed along various roads in Taiwan. Image data

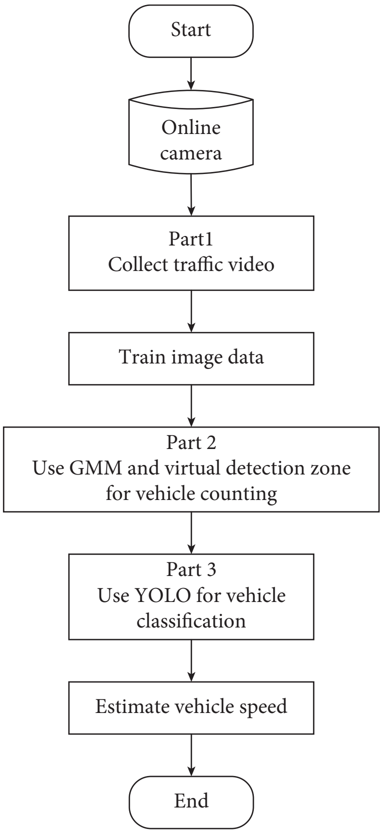
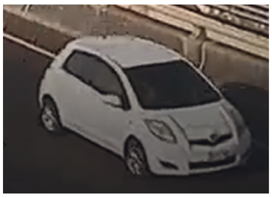
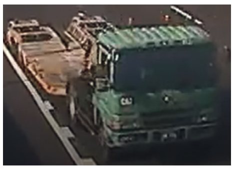
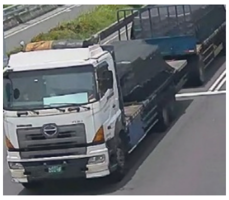


Figure 1: Flowchart of the vehicle counting and classification process.

Table 1: Vehicle classification.

Class	Vehicle	Length (m)	Image
1	Sedan	3.6–5.5	
2	Truck	>5.5–11	
3	Scooter	1–2.5	
4	Bus	7–12.2	
5	Hlinkcar	15–18	
6	Flinkcar	18–20	

were extracted from the traffic videos using a script, and labeling was performed using an open-source software application called "labeling" [25]. According to the common types of vehicles on the road are announced by the Directorate General of Highways, Ministry of Transportation and Communications (MOTC) in Taiwan, this study divides six different sizes, such as sedans, trucks, scooters, buses, hlinkcars, and flinkcars, in the training process, and the vehicle lengths of these six vehicle classes are listed in Table 1. In this study, we used YOLO to perform vehicle classification without using the length of the vehicle.

2.2. Vehicle Counting. To count vehicles, a GMM is used for the background subtraction in the complex environment to identify the regions of moving objects. The GMM is quite reliable in the background extraction and foreground segmentation process, so the characteristics of a moving object in video surveillance are easier to detect [26, 27]. The virtual detection zone is predefined in each video and used for vehicle feature computation. When the vehicle enters a virtual detection zone and virtual detection lane line, the GMM is used for vehicle counting. The vehicle counting window is depicted in Figure 2.

2.3. Vehicle Detection and Classification. This study uses the YOLO algorithm to classify vehicles into six classes. The validation method is used for verifying the vehicle classification in the collected videos. A visual classifier based on the YOLO algorithm is used to verify the vehicle classification capability. Figure 3 depicts the architecture of the visual classifier based on the YOLO algorithm that is used for classifying each vehicle into one of six classes. In the training process, when a vehicle belonging to one of the six classes is detected, all bounding boxes are extracted, their classes are manually labeled, and the labeled data are passed to the YOLO model for classifying the vehicle.

The YOLOv3 model architecture displayed in Figure 4 was used in this study. Images of size 416 × 416 px were input into the Darknet-53 network. This feature extraction network comprises 53 convolutional layers, and thus, it is called

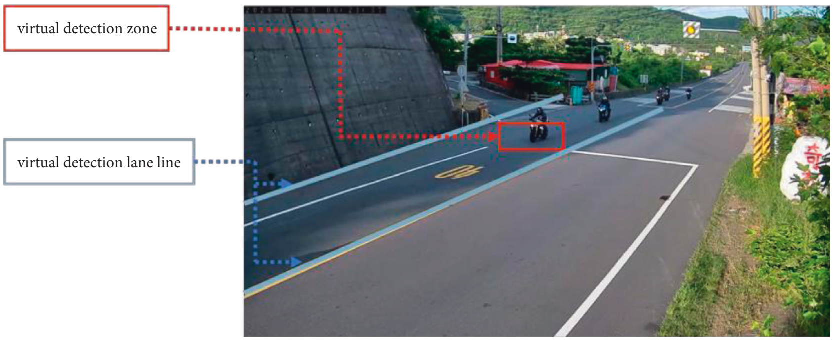


Figure 2: Object detection window.

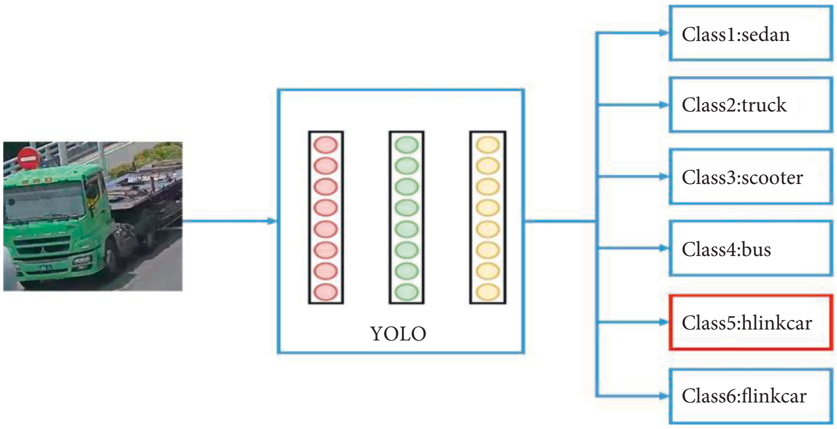


Figure 3: Architecture of visual classifier based on the YOLO algorithm for verifying the vehicle classification.

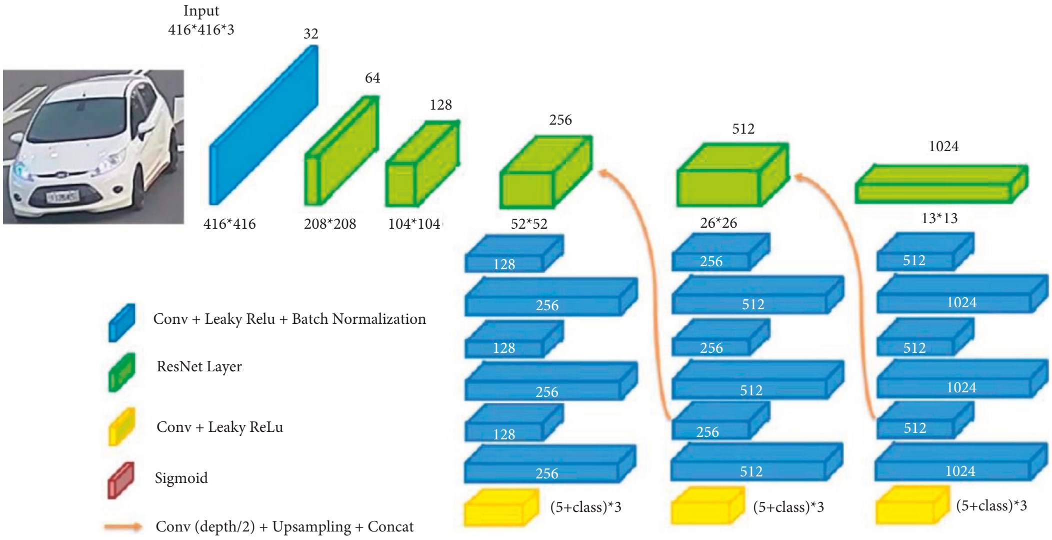


Figure 4: YOLOv3 model architecture.

Darknet-53 [11]. In Darknet-53, alternating convolution kernels are used, and after each convolution layer, a batch normalization layer is used for normalization. The leaky rectified linear unit function is used as the activation function, the pooling layer is discarded, and the step size of the convolution kernel is increased to reduce the size of the feature map. The YOLOv3 model uses ResNet for feature extraction and subsequently uses the feature pyramid top-down and lateral connections to generate three features with sizes of $13 \times 13 \times 1024$, $26 \times 26 \times 512$, and $52 \times 52 \times 256$ px. The final output depth is $(5 + \text{class}) \times 3$, which indicates that the following parameters are predicted: four basic parameters and the credibility of a box across three regression bounding boxes as well as the possibility of each class being contained in the bounding box. YOLOv3 uses the sigmoid function to score each class. When the class score is higher than the threshold, the object is considered to belong to a given category, and any object can simultaneously have multiple class identities without conflict.

The loss function of YOLOv3 is mainly divided into four parts. *A* denotes the loss of the identified center coordinates that is used to predict (x, y) in the bounding box to ensure that it is only valid for the highest predicted target. *B* is the loss of (w, h) width and height in the predicted bounding box, and the error value reflects the bounding box of different sizes in the object to predict the square root of the width and height instead of directly predicting the width and height of the bounding box. *C* is the loss of the predicted object category, assuming that each box is a cell; if the center of the object detection is in this cell, then mark the cell with bounding box (x, y, w, h), and there is also category information to meet which object in the image to predict in the cell. *D* denotes the loss of the credibility of the predicted object to calculate the credibility in each bounding box to know that when the bounding box predicts the object. When the object is not predicted, there will be a credibility prediction penalty $\lambda_{\text{noobj}} = 0.5$, and it is defined as follows:

$$
\begin{aligned}
&\overbrace{\lambda_{coord}\sum_{i=0}^{s^2}\sum_{j=0}^{B}\ell_{ij}^{obj}\left[(x_i-\hat{x}_i)^2+(y_i-(\hat{y}_i))^2\right]}^{A}\\
&\overbrace{+\lambda_{coord}\sum_{i=0}^{s^2}\sum_{j=0}^{B}\ell_{ij}^{obj}\left[\left(\sqrt{w_i}-\sqrt{\hat{w}_i}\right)^2+\sqrt{h_i}-\sqrt{\hat{h}_i}\right)^2\right]}^{B}\\
&\overbrace{+\sum_{i=0}^{s^2}\ell_{i}^{obj}\sum_{j=0}^{B}\left[(p_i(c)-\hat{p}_i(c))^2\right]}^{C}\\
&\overbrace{+\sum_{i=0}^{s^2}\sum_{j=0}^{B}\ell_{ij}^{obj}\left[\left(C_i-\hat{C}_I\right)^2\right]+\lambda_{noobj}\sum_{i=0}^{s^2}\sum_{j=0}^{B}\ell_{ij}^{noobj}\left[\left(C_i-\hat{C}_I\right)^2\right]}^{D},
\end{aligned}
\tag{1}
$$

where x_i, y_i is the location of the centroid of the anchor box and w_i, h_i is the width and height of the anchor box. C_i is the *Objectness*, i.e., confidence score of whether there is an object or not, and $p_i(c)$ is the classification loss.

YOLOv4 is the latest algorithm of YOLO series, which is the basis of YOLOv3, scales both up and down and is applicable to small and large networks while maintaining optimal speed and accuracy, and the network architecture is shown in Figure 5. Compared with YOLOv3, YOLOv4-tiny is an extended version of YOLOv3. The original Darknet53 network is added with a CSP network. Backbone is CSPOSANet proposed by Cross Stage Partial Network (CSPNet) + One-Shot Aggregation Network (OSANet), plus Partial in Computational Blocks (PCB) technology. CSPNet can be applied to different CNN architectures to reduce the amount of parameters and calculations while improving accuracy. OSANet is derived from the OSA model in VoVNet. Its central idea is improved by the DenseNet module. At the end, all layers are connected to allow input consistent with the number of output channels; PCB technology can make the model more flexible because it can be adjusted according to the structure to achieve the best accuracy-speed balance.

The loss function remains the same as the YOLOv4 model, which consists of three parts: classification loss, regression loss, and confidence loss [28]. Classification loss and confidence loss remain the same as the YOLOv3 model, but complete intersection over union (CIoU) is used to replace mean-squared error (MSE) to optimize the regression loss [29]. The CIoU loss function is shown as follows:

$$
\begin{aligned}
\text{LOSS} = {} & 1-\text{IoU}+\frac{\rho^2\left(b,b^{gt}\right)}{c^2}+\alpha v\\
&-\sum_{i=0}^{S^2}\sum_{j=0}^{B}I_{ij}^{\text{obj}}\left[\hat{C}_i\log\left(C_i\right)+\left(1-\hat{C}_i\right)\log\left(1-C_i\right)\right]\\
&-\lambda_{\text{noobj}}\sum_{i=0}^{S^2}\sum_{j=0}^{B}I_{ij}^{\text{noobj}}\left[\hat{C}_i\log\left(C_i\right)+\left(1-\hat{C}_i\right)\log\left(1-C_i\right)\right]\\
&-\sum_{i=0}^{S^2}I_{ij}^{\text{obj}}\sum_{c\in\text{classes}}\left[\hat{p}_i(c)\log\left(p_i(c)\right)+\left(1-\hat{p}_i(c)\right)\log\left(1-p_i(c)\right)\right],
\end{aligned}
\tag{2}
$$

where S^2 represents $S\times S$ grids; each grid generates B candidate boxes, and each candidate box gets corresponding bounding boxes through the network; finally, $S\times S\times B$ bounding boxes are formed. If there is no object (noobj) in the box, only the confidence loss of the box is calculated. The confidence loss function uses cross entropy error and is divided into two parts: there is the object (obj) and noobj. The loss of noobj increases the weight coefficient λ, which is to reduce the contribution weight of the noobj calculation part. The classification loss function also uses cross entropy error. When the j-th anchor box of the i-th grid is responsible for certain ground truth, then the bounding box generated by this anchor box will calculate the classification loss function.

2.4. Speed Estimation.

The real-time vehicle speed is also calculated in this study. Figure 6 shows the video images taken along the direction parallel to the length of the car (defined as the y-axis) and parallel to the width of the car (defined as the x-axis). First, as per the scale in the video, the yellow line (referred to as L) in the red circle has a length of 4 m in accordance with traffic laws. A GMM is used to draw a virtual detection zone (blue box) on the road to be tested (referred to as Q). The green box is the car frame (referred to as C), and the midpoint of the car is Ct.

$$
\begin{aligned}
u_0 &= \frac{\overline{L_{\text{AB}}}(px)}{\overline{L_{\text{AB}}}(m)},\\
\alpha &= \frac{u_0}{u_0'},
\end{aligned}
\tag{3}
$$

$$
\text{If } \alpha>1,\quad u_i=u_0\alpha^i, u_j=\frac{u_0}{\alpha^j},
\tag{4}
$$

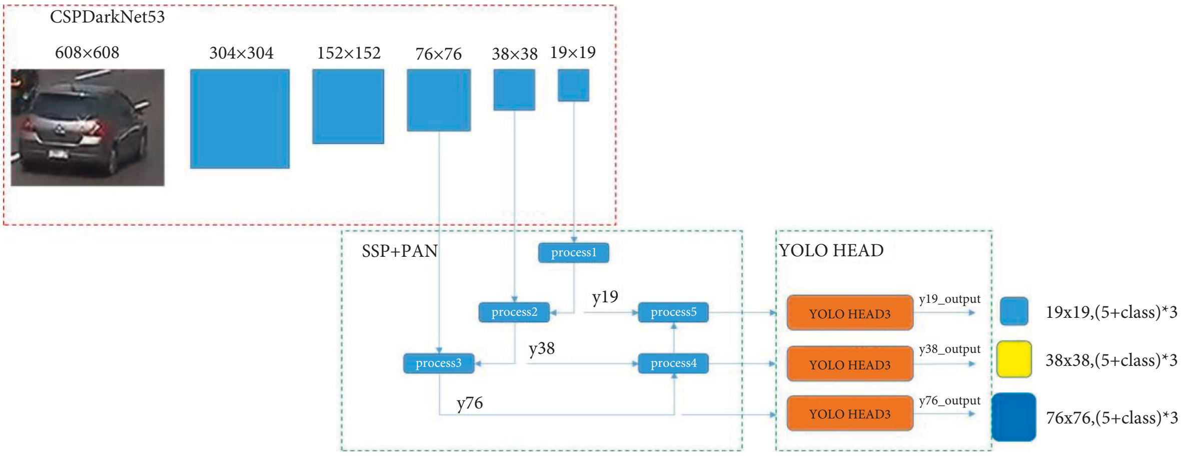


Figure 5: YOLOv4 model architecture.

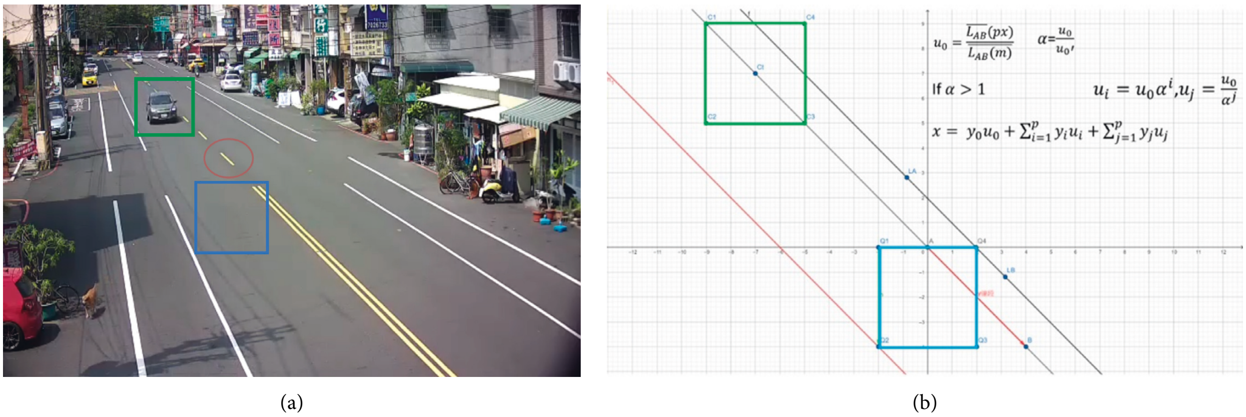

(a)
(b)
$u_0 = \frac{\overline{L_{AB}}(px)}{\overline{L_{AB}}(m)}$
$\alpha = \frac{u_0}{u_0'}$
If $\alpha > 1$
$u_i = u_0\alpha^i, u_j = \frac{u_0}{\alpha^j}$
$x = y_0u_0 + \sum_{i=1}^{p} y_iu_i + \sum_{j=1}^{p} y_ju_j$
LA
LB
A
B
Q1
Q2
Q3
Q4
C1
C2
C3
C4
Ct

Figure 6: Diagram of speed estimation ((a): real picture on the video; (b): control scale).

where u_0 is the scale, i is the scale of the blue box, j is the scale of the green box, px is the length of the video, and m is the actual length. The parameter α denotes the increase or decrease in relationship of the scale per unit length on the y-axis. If $\alpha > 1$, the speed calculation is performed using equation (4).

To calculate the parallel L line segment of Ct (referred to as L^*), the algorithm computes the L^* distance y between A and B. Then, it restores y from its scale relationship with the actual line segment length x, where x denotes the distance traveled by the vehicle in Q.

$$x = y_0u_0 + \sum_{i=1}^{p} y_iu_i \sum_{j=1}^{p} y_ju_j. \tag{5}$$

In the calculation process, the program is used to determine the frame rate of the video and calculate the number of frames for which the vehicle travels in Q (referred to as p). Equation (6) is used to find the travel time of the vehicle from A to B in Q.

$$t = \frac{p}{\text{fps}}, \tag{6}$$

$$v = \frac{x}{t}, \tag{7}$$

$$v' = x \times \frac{3.6}{t}. \tag{8}$$

Equation (7) is used for calculating vehicle speed. After unit conversion (m/s to km/h), Equation (8) provides the vehicle speed.

3. Results and Discussion

All experiments in this study were performed using the YOLO algorithm under the Darknet framework, and the program was written in *Python* 2.7. To validate the real-time traffic monitoring system, we used a real-world data set to perform vehicle detection, vehicle counting, speed estimation, and classification. In this study, three test data sets were used to evaluate the proposed method. One of these data sets was mainly derived from traffic video images of online cameras on various roads in Taiwan, and it contains 12,761 training images and 3,190 testing images. Second, the Montevideo Audio and Video Dataset (MAVD), which contains data on different levels of traffic activity and social use characteristics in Montevideo city, Uruguay, was used as the other traffic data set [30]. Finally, GARM Road-Traffic Monitoring (GRAM-RTM) data set [21] has four categories (i.e., cars, trucks, vans, and big-trucks). The total number of different objects in each sequence is 256 for M-30, 235 for M-30-HD, and 237 for Urban 1. In this study, the definition of accuracy is based on the classification of vehicles in the database. In the video verification, if the results of manual and proposed system classification of the vehicles are the same, it means that the count is correct; otherwise, it is the wrong vehicle counting.

3.1. Vehicle Counting.

3.1. Vehicle Counting. Seven input videos of the road, each ranging in length between 3 and 5 minutes, were recorded at 10 am and 8 pm. In addition, eleven input videos of the road in the rain were also recorded for testing. Each frame in these traffic videos was captured at 30 fps. The first experimental results of real-time vehicle counting using the proposed method during the day are summarized in Table 2. The symbols S and L denote small and large vehicles, respectively. The vehicle counting accuracy of the proposed method at 10 am was 95.5%. The second experimental results of real-time vehicle counting using the proposed method during the night are summarized in Table 3. The vehicle counting accuracy of the proposed method at 8 pm was 98.5%. In addition, the third experimental results of real-time vehicle counting using the proposed method in the rain are summarized in Table 4. The vehicle counting accuracy of the proposed method was 94%. Screenshots of vehicle detection with the proposed real-time vehicle counting and classification system are depicted in Figure 7, where the detected vehicles are represented as green rectangles.

Vehicle counting in online videos is delayed due to network stoppages or because the target vehicle may be blocked by other vehicles on the screen, which causes the count to be missed. In addition, poor lighting in the rain and night affects the vehicle recognition capabilities of YOLOv3 and YOLOv4. These challenges can be overcome using a stable network connection and adjusting the camera brightness, respectively. Therefore, the novelty of this study is to solve the problem of unclear recognition in the rain.

3.2. Speed Estimation. In this subsection, the vehicle speed can be estimated using the proposed method. Table 5 lists the actual and the estimated speeds of the vehicles. The results indicate that the average absolute percentage error of vehicle speed estimation was about 7.6%. The use of online video for vehicle speed estimation will cause large speed errors due to network delays. Therefore, network stability is essential to reduce the percentage error in the speed estimation.

3.3. Comparison Results Using the MAVD and GRAM-RTM Data Sets. MAVD traffic data set [30] and GARM Road-Traffic Monitoring (GRAM-RTM) data set [21] were used for evaluating the vehicle counting performance of the proposed method. The videos were recorded with a GoPro Hero 3 camera at a frame rate of 30 fps and a resolution of 1920 × 1080 px. We analyzed 10 videos, and the vehicle counting accuracy of the proposed method at 10 am for the MAVD traffic data set was 93.84%. Vehicle classification results of the proposed method using MAVD traffic data set are listed in Table 6.

Table 2: The real-time vehicle counting using the proposed method in the daytime.

Video no.	Actual number of vehicles			Estimated number of vehicles		
	S	L	Total	S	L	Total
1	31	7	38	29	6	35
2	18	0	18	18	0	18
3	16	5	21	16	5	21
4	28	0	28	25	0	25
5	22	3	25	20	3	23
6	11	0	11	11	0	11
7	11	1	12	9	0	9

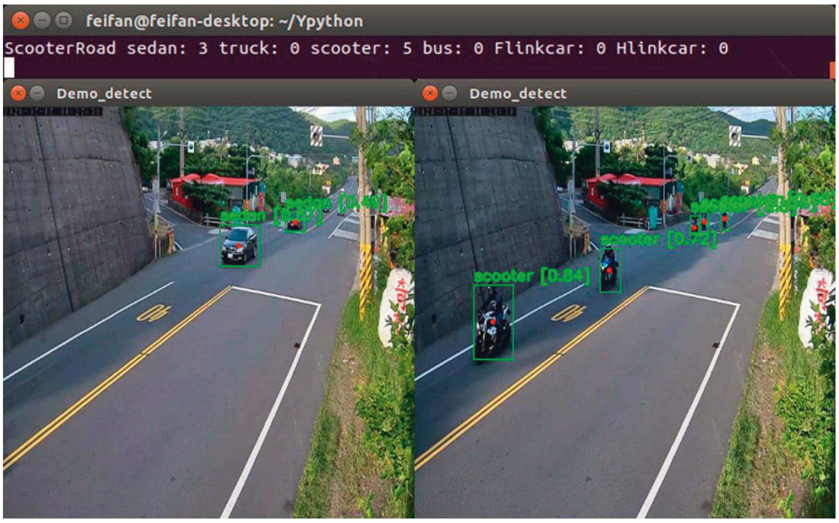
Table 3: The real-time vehicle counting using the proposed method in the night time.

Video no.	Actual number of vehicles			Estimated number of vehicles		
	S	L	Total	S	L	Total
1	23	8	31	23	7	30
2	31	5	36	29	5	34
3	15	2	17	14	1	15
4	10	3	13	9	3	12
5	19	4	23	19	4	23
6	11	2	13	11	2	13
7	35	5	38	34	3	37

Table 4: The real-time vehicle counting using the proposed method in the raining day.

Video no.	Actual number of vehicles			Estimated number of vehicles		
	S	L	Total	S	L	Total
1	9	0	9	9	0	9
2	12	0	12	10	1	11
3	13	0	13	13	0	13
4	7	0	7	8	0	8
5	11	1	12	14	1	15
6	15	0	15	17	0	17
7	10	0	10	13	0	13
8	7	0	7	10	0	10
9	12	0	12	15	0	15
10	12	0	12	14	0	14
11	17	0	17	19	0	19

In summary, three data sets, namely, MAVD, GRAM-RTM, and our collection data sets, were used to verify the proposed method and Fast RCNN method [10]. The MAVD training and testing samples contains vehicles belonging to four categories (i.e., cars, buses, motorcycles, and trucks). The GRAM-RTM data set has four categories (i.e., cars, trucks, vans, and big-trucks). The total number of different objects in each sequence is as follows: 256 for M-30, 235 for M-30-HD, and 237 for Urban 1. Table 7 shows the classification accuracy results of three data sets using various methods. In Table 7, the



```
ScooterRoad sedan: 3 truck: 0 scooter: 5 bus: 0 Flinkcar: 0 Hlinkcar: 0
```

Figure 7: Screenshots from the proposed real-time vehicle counting, speed estimation, and classification system.

Table 5: The actual and the estimated vehicle speeds using the proposed method.

Vehicle ID	Actual speed	Estimated speed	Difference	Error (%)
1	60	63	3	5
2	70	75	5	7
3	72	63	−9	12.5
4	99	100	1	1
5	84	85	1	1
6	67	60	−7	10
7	73	71	−2	2.7
8	67	64	−3	4.4
9	37	43	6	16
10	73	77	4	5
11	55	50	−5	9
12	48	54	6	12.5
13	111	127	16	14.4
14	79	75	−4	5
15	69	71	2	2.8
16	82	75	−7	8.5
17	83	73	−10	12
	Average error			**7.6**

Table 6: Vehicle classification results of the proposed method using MAVD traffic data set.

Video no.	Total number of vehicles			Number of counted vehicles		
	S	*L*	Total	*S*	*L*	Total
1	8	0	8	8	0	8
2	5	0	5	4	0	4
3	4	2	6	3	2	5
4	4	0	4	3	0	3
5	3	0	3	3	1	4
6	1	0	1	1	0	1
7	7	2	9	7	2	9
8	11	0	11	10	0	10
9	9	0	9	9	0	9
10	9	0	9	8	0	8

proposed method with YOLOv4 achieved the highest classification accuracy of 98.91% and 99.5% in MAVD and GRAM-RTM data sets, respectively. Moreover, three different environments (i.e., daytime, night time, and rainy day) are used verify the proposed method. Experimental results indicate that the proposed method with YOLOv4 also achieves the highest classification accuracy of 99.1%, 98.6%, and 98% in daytime, night time, and rainy day, respectively.

Recently, some researchers have adopted various methods for vehicle classification using GRAM-RTM data set, such as Faster RCNN [10], CNN [31], and DNN [32]. Therefore, we use the same GRAM-RTM data set to compare the proposed method with other methods. Table 8 shows the comparison results. In Table 8, the results show that the proposed method with YOLOv4 can perform better than the other methods.

Table 7: Classification accuracy results of three data sets using various methods.

Data sets		Methods	Accuracy (%)	FPS
MAVD		Faster RCNN [10]	97.21	5
		Proposed method with YOLOv3	97.66	15
		Proposed method with YOLOv4	98.91	15
GRAM-RTM		Faster RCNN [10]	91.54	5
		Proposed method with YOLOv3	98.02	15
		Proposed method with YOLOv4	99.5	15
Our data set	Daytime	Faster RCNN [10]	97.7	5
		Proposed method with YOLOv3	98	15
		Proposed method with YOLOv4	99.1	15
	Night time	Faster RCNN [10]	93.59	5
		Proposed method with YOLOv3	98	15
		Proposed method with YOLOv4	98.6	15
	Rainy day	Faster RCNN [10]	87.5	5
		Proposed method with YOLOv3	90	15
		Proposed method with YOLOv4	98	15

Table 8: Classification accuracy results of various methods using GARM-RTM data set.

Methods	Faster RCNN [10]	Gomaa et al. [31]	Abdelwahab [32]	Our proposed method	
				YOLOv3	YOLOv4
Accuracy (%)	91.54	96.8	93.51	98.02	99.5

4. Conclusions

In this study, a real-time traffic monitoring system based on a virtual detection zone, GMM, and YOLO is proposed for increasing the vehicle counting and classification efficiency. GMM and a virtual detection zone are used for vehicle counting, and YOLO is used to classify vehicles. Moreover, the distance and time traveled by a vehicle are used to estimate the speed of the vehicle. In this study, MAVD, GRAM-RTM, and our collection data sets are used to verify the proposed method. Experimental results indicate that the proposed method with YOLOv4 achieved the highest classification accuracy of 98.91% and 99.5% in MAVD and GRAM-RTM data sets, respectively. Moreover, the proposed method with YOLOv4 also achieves the highest classification accuracy of 99.1%, 98.6%, and 98% in daytime, night time, and rainy day, respectively. In addition, the average absolute percentage error of vehicle speed estimation with the proposed method is about 7.6%. Therefore, the proposed method can be applied to vehicle counting, speed estimation, and classification in real time.

However, the proposed method has a few limitations. The vehicles appearing in the video are assumed to be inside the virtual detection zone; thus, the width of the virtual detection zone should be sufficiently large for counting the vehicles. In the future work, we will focus on algorithm acceleration and model simplification.

Data Availability

The MAVD and GRAM-RTM traffic data sets are available at https://zenodo.org/record/3338727#.YBD8B-gzY2w and https://gram.web.uah.es/data/datasets/rtm/index.html.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this study.

Acknowledgments

This research was funded by the Ministry of Science and Technology of the Republic of China, grant number MOST 110-2221-E-167-031-MY2.

References

[1] Y. Mo, G. Han, H. Zhang, X. Xu, and W. Qu, "Highlight-assisted nighttime vehicle detection using a multi-level fusion network and label hierarchy," *Neurocomputing*, vol. 355, pp. 13–23, 2019.

[2] D. Feng, C. Haase-Schuetz, L. Rosenbaum et al., "Deep multi-modal object detection and semantic segmentation for autonomous driving: datasets, methods, and challenges," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 3, p. 3, 2019.

[3] Z. Liu, Y. Cai, H. Wang et al., "Robust target recognition and tracking of self-driving cars with radar and camera information fusion under severe weather conditions," *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[4] Y. Qian, J. M. Dolan, and M. Yang, "DLT-NET: joint detection of drivable areas, lane lines, and traffic objects," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 11, pp. 4670–4679, 2020.

[5] Y. Cai, L. Dai, H. Wang et al., "Pedestrian motion trajectory prediction in intelligent driving from far shot first-person perspective video," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–16, 2021.

[6] Á. Llamazares, E. J. Molinos, and M. Ocaña, "Detection and tracking of moving obstacles (DATMO): a review," *Robotica*, vol. 38, no. 5, pp. 761–774, 2020.

[7] C. Liu, D. Q. Huynh, Y. Sun, M. Reynolds, and S. Atkinson, "A vision-based pipeline for vehicle counting, speed estimation, and classification," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–14, 2020.

[8] Y.-Q. Huang, J.-C. Zheng, S.-D. Sun, C.-F. Yang, and J. Liu, "Optimized YOLOv3 algorithm and its application in traffic flow detections," *Applied Sciences*, vol. 10, Article ID 3079, 2020.

[9] W. Liu, D. Anguelov, D. Erhan et al., "SSD: single shot multibox detector," in *Proceedings of the Computer Vision—ECCV 2016*, pp. 21–37, Amsterdam, Netherlands, October 2016.

[10] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.

[11] X. Zhang and X. Zhu, "Vehicle detection in the aerial infrared images via an improved Yolov3 network," in *Proceedings of the 2019 IEEE 4th International Conference on Signal and Image Processing (ICSIP)*, pp. 372–376, Wuxi, China, July 2019.

[12] A. Bochkovskiy, C. Y. Wang, and H. Y. M. Liao, "YOLOv4: optimal speed and accuracy of object detection," 2020, https://arxiv.org/abs/2004.10934v1.

[13] J. Redmon and A. Farhadi, "YOLO V3: an incremental improvement," pp. 1–22, 2018, http://arxiv.org/abs/1804.02767.

[14] D. Biswas, H. Su, C. Wang, A. Stevanovic, and W. Wang, "An automatic traffic density estimation using single shot detection (SSD) and mobilenet-SSD," *Physics and Chemistry of the Earth, Parts A/B/C*, vol. 110, pp. 176–184, 2019.

[15] W. Yang, Z. Li, C. Wang, and J. Li, "A multi-task Faster R-CNN method for 3D vehicle detection based on a single image," *Applied Soft Computing*, vol. 95, Article ID 106533, 2020.

[16] X. Hu, Z. Wei, and W. Zhou, "A video streaming vehicle detection algorithm based on YOLOv4," in *Proceedings of the 2021 IEEE 5th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, pp. 2081–2086, Chongqing, China, March 2021.

[17] C. Y. Cao, J. C. Zheng, Y. Q. Huang, J. Liu, and C. F. Yang, "Investigation of a promoted You Only Look once algorithm and its application in traffic flow monitoring," *Applied Sciences*, vol. 9, Article ID 3619, 2019.

[18] H. Zhou, L. Wei, C. P. Lim, D. Creighton, and S. Nahavandi, "Robust vehicle detection in aerial images using bag-of-words and orientation aware scanning," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 12, pp. 7074–7085, 2018.

[19] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 936–944, Honolulu, HI, USA, July 2017.

[20] C.-Y. Chen, Y.-M. Liang, and S.-W. Chen, "Vehicle classification and counting system," in *Proceedings of the 2014 International Conference on Audio, Language and Image Processing (ICALIP)*, pp. 485–490, Shanghai, China, July 2014.

[21] N. Seenouvong, U. Watchareeruetai, C. Nuthong, K. Khongsomboon, and N. Ohnishi, "Vehicle detection and classification system based on virtual detection zone," in *Proceedings of the 2016 13th International Joint Conference on Computer Science and Software Engineering (JCSSE)*, Khon Kaen, Thailand, July 2016.

[22] A. Grents, V. Varkentin, and N. Goryaev, "Determining vehicle speed based on video using convolutional neural network," *Transportation Research Procedia*, vol. 50, pp. 192–200, 2020.

[23] S. Tabassum, M. S. Ullah, N. H. Al-Nur, and S. Shatabda, "Native vehicles classification on Bangladeshi roads using CNN with transfer learning," in *Proceedings of the 2020 IEEE Region 10 Symposium (TENSYMP)*, Dhaka, Bangladesh, June 2020.

[24] S. Tabassum, S. Ullah, N. H. Al-nur, and S. Shatabda, "Poribohon-BD: Bangladeshi local vehicle image dataset with annotation for classification," *Data in Brief*, vol. 33, Article ID 106465, 2020.

[25] LabelImg (accessed on 5 March 2018), https://github.com/tzutalin/labelImg.

[26] A. Nurhadiyatna, B. Hardjono, A. Wibisono et al., "Improved vehicle speed estimation using Gaussian mixture model and hole filling algorithm," in *Proceedings of the 2013 International Conference on Advanced Computer Science and Information Systems (ICACSIS)*, Sanur Bali, Indonesia, September 2013.

[27] A. Ghosh, M. S. Sabuj, H. H. Sonet, S. Shatabda, and D. M. Farid, "An adaptive video-based vehicle detection, classification, counting, and speed-measurement system for real-time traffic data collection," in *Proceedings of the 2019 IEEE Region 10 Symposium (TENSYMP)*, Olkata, India, June 2019.

[28] L. Wu, J. Ma, Y. Zhao, and H. Liu, "Apple detection in complex scene using the improved YOLOv4 model," *Agronomy*, vol. 11, no. 3, p. 476, 2021.

[29] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, and D. Ren, "Distance-IoU loss: faster and better learning for bounding box regression," in *Proceedings of the 2020 AAAI Conference on Artificial Intelligence*, pp. 12993–13000, New York, NY, USA, February 2020.

[30] P. Zinemanas, P. Cancela, and M. Rocamora, "MAVD: a dataset for sound event detection in urban environments," in *Proceedings of the 4th Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE 2019)*, New York, NY, USA, October 2019.

[31] A. Gomaa, M. M. Abdelwahab, M. Abo-Zahhad, T. Minematsu, and R.-I. Taniguchi, "Robust vehicle detection and counting algorithm employing a convolution neural network and optical flow," *Sensors*, vol. 19, no. 20, Article ID 4588, 2019.

[32] M. A. Abdelwahab, "Accurate vehicle counting approach based on deep neural networks," in *Proceedings of the 2019 International Conference on Innovative Trends in Computer Engineering (ITCE'2019)*, Aswan, Egypt, February 2019.