International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 7, Issue 11, pp 378-384, November-2023

https://doi.org/10.47001/IRJIET/2023.711051

Automated Traffic Law Enforcement System

[1]Omesh Diyamantha, [2]Milinda Hewavitharana, [3]Rehan Perera

[1,2,3]Dept. of Computer Systems Engineering, Sri Lanka Institute of Information Technology, Malabe 10115, Sri Lanka
Authors E-mail: [1]it18223118@my.sliit.lk, [2]it20501402@my.sliit.lk, [3]it19029696@my.sliit.lk

***Abstract* - Road accidents and traffic offenses hinder growth and cost Sri Lanka money. From 2016 to June 2023, 223,000 accidents killed over 20,000 people, 8 every day. Road fatalities are now over 120 per million, greater than the US and Japan. Rapid motorization without infrastructure growth, lax enforcement, absence of speeding and drunk driving punishments, and defective violation reporting systems are important issues. An Automated red Violation Detection, Reporting, and Fine System uses computer vision, deep learning, and IoT to identify speeding, lane breaches, and red-light disobedience via video cameras. It would immediately report offenses and publicly fine without traffic police. Vehicle detection, speed identification, lane infraction recognition, and traffic light classification are accurate under real-world scenarios thanks to Deep Learning models. Over 90% speed violation detection accuracy in diverse weather is achieved with extensive data augmentation. Through 24/7 monitoring and transparency, automated systems deter noncompliance, reduce accidents, and improve road safety. Results show that traffic video insights can be used to construct intelligent law enforcement systems.**

Keywords: Image Processing, YOLO, Lane Violation, Speed Violation, Red Light Violation, ANPR, Deep Learning.

I. INTRODUCTION

Sri Lanka’s Road accident and traffic violation crisis hinders sustainable growth and costs money. Between 2016 and June 2023, 223,000 incidents killed almost 20,000 people, 8 every day [1]. In 2022, 2,371 of 19,740 road accidents in SriLanka were fatal, according to traffic police data. The victims include 792 pedestrians, 820 motorcyclists, 189 drivers, 314 passengers, 226 cyclists, 189 rear riders, and 06 others. There were 14,579 injuries, 6,264 serious injuries, and 9,365 minor injuries [2]. Traffic accidents in Sri Lanka have hampered national growth and caused enormous economic losses. Rapid motorization, lax discipline, enforcement, and reporting systems cause high traffic accident mortality. An Automated Traffic Violation Detection and Reporting System uses computer vision and deep learning to detect violations via cameras and fine violators without police.

The system uses precise vehicle detection, speed identification, lane infraction recognition, and traffic light classification models. Significant data augmentation improves speed violation detection in various weather conditions. Automated 24/7 monitoring improves safety, transparency, and accident reduction. Video quality, occlusion, and training data size are issues. However, the solution improves discipline, deterrent, and safety over manual enforcement. Sri Lanka can improve road safety using intelligent transportation systems. The methods are ready for use after testing. The automated system uses computer vision and deep learning to analyze footage, automate enforcement, prevent accidents, and promote sustainable growth. Ease of Use

II. BACKGROUND AND LITERATURE SURVEY

Modern road and traffic management is essential to eliminating these fatal accidents and the traffic offences that cause most road fatalities. Thus, an ANPR system that can identify the vehicle's owner to issue penalties, handle fines, and resolve traffic infraction issues is necessary. Peiris, Akila Edirisuriya, et al. suggested an automatic real-time traffic infraction detection system for congested highways with lawbreakers. The suggested system detects traffic violations using computer vision and machine learning [3]. A Alaydrus, W.K Putra, et al. developed an RFID-based traffic violation detection system. This system identifies low-power RFID-tagged vehicles. Computer vision uses camera sensors for image and video processing like license plate recognition [4]. Mohammed Imran Basheer Ahmed, Rim Zaghdoud, Mohammed Salih Ahmed, et al. created a computer vision-based system to detect and warn of traffic accidents in real time. The suggested framework has three models. Each model is built in a prototype user interface [5] to analyze the system's structure. Cheon et al. developed another vision-based vehicle identification method. SVM and Histogram of Oriented Gradients (HOG) features were employed to classify pictures by automobile content [6]. Dubai's RTA boasts a cutting-edge traffic management system. This system uses radar with a camera, sensors, transmitter-receiver, data server, and other electronics. Its main application is to generate fines accompanying visual evidence of transgressions such license plate recognition, speeding, illegal U-turns, and unauthorized cars. Lots of overlap in this study. A specific combination of two cameras was used for accessible technology. The high-

© 2023-2017 IRJIET All Rights Reserved

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 7, Issue 11, pp 378-384, November-2023

https://doi.org/10.47001/IRJIET/2023.711051

angle placement of one camera made the offense obvious, while the other was employed for something different. The best vehicle photography requires an off-angle view. Not being done in real time is the main difference. Implementing this plan is next [7]. Sri Lanka needs a reliable, accurate, and affordable traffic violation system, as shown. We suggest increasing fines for moving offenses such violating traffic lights, weaving between lanes, and speeding to decrease deadly car accidents. The study also examines the system's performance in fog, rain, and sunshine.

III. PROPOSED METHODOLOGY

Designing an intelligent traffic system to automatically detect red light running, speeding, and unlawful lane changes is the goal. Without human intervention, it finds offending automobiles, retrieves licenses, and generates digital fines. Design emphasizes:

Violation Detection Accuracy: Traffic offenses are detected by deep learning systems. They identify red light running, speeding, and unlawful lane changes using real-time video data. Deep learning models are trained on diverse data for correct performance in various scenarios.

Edge Processing for Low Latency: At the edge, near traffic cameras, the system processes data to reduce response times. This decreases violation detection latency and speeds choices. Edge computing ensures network stability during disturbances. Intuitive Web Dashboard: Authorities like traffic police can use the system's intuitive web dashboard. Real-time infraction information enables swift, educated actions. Video playback, violation data, and digital fines ease administration.

3.1 System Architecture

Edge Devices: Compact camera PCs are placed at intersections. These edge devices use deep learning to swiftly discover infractions in local video. Only violation events are processed in the cloud. Edge devices lower bandwidth and latency. Servers: Central servers collect edge device violations. Vehicle data is recorded. Technology simplifies penalties by generating electronic challans for proved infractions. Scalability and centralized data management are achievable with this technology.

Web Application: A user-friendly web software lets traffic officials evaluate violation incidents and footage. It handles e-challans. Analytics capabilities to track traffic and violations could be added.

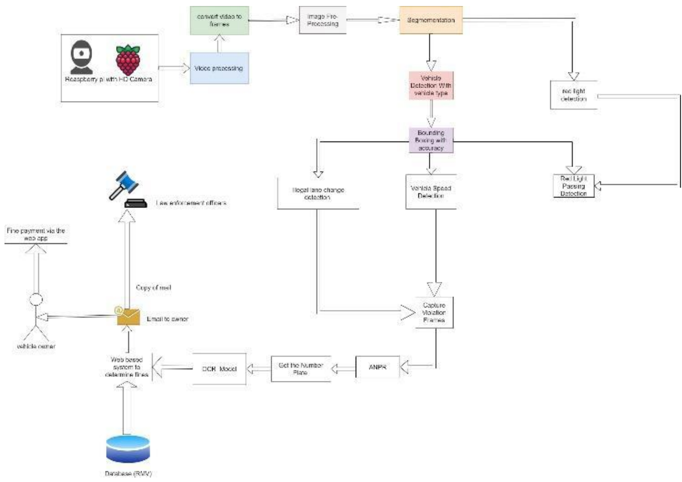


Figure 1: System Architecture

Deep learning video analytics for low-latency traffic control relies on edge devices. They can compute and process data at the network edge in under 100ms. Edge devices detect violations and relay selective events for traffic surveillance, reducing bandwidth. Traffic officials can get real-time insights via a web app. The system's efficiency depends on deep learning models that analyze footage to find violations. Real-time infraction detection by edge devices and selective data transmission improves road safety and efficiency.

3.2 Technologies used to build the system

System backend and primary processing were implemented in Python. Computer vision operations including frame loading and preprocessing using OpenCV. Keras was used to define and train the neural network and augment images. SciKit Learn partitioned the dataset, while Numpy processed frame data. Local web server made with Flask. MongoDB enabled remote infraction storage.

3.3 Deep Learning Models

Deep learning models enhance traffic safety and efficiency. Different neural networks address speeding, red light running, and unlawful lane changes. For efficiency, they're adapted to local traffic. Models learn region-specific characteristics and properly identify violations using local camera datasets, enhancing road safety and enforcement. Deep learning adjusts to local variables, helping authorities understand traffic infractions.

3.3.1 Lane Violation Detection Model

The 2000 30-fps video clip dataset includes various vehicles, lighting, and weather. Training uses 80%, validation 20% for rigorous evaluation. Cropping and noise improve robustness. A 16-batch Tesla T4 GPU trains for 12 hours. The

© 2023-2017 IRJIET All Rights Reserved

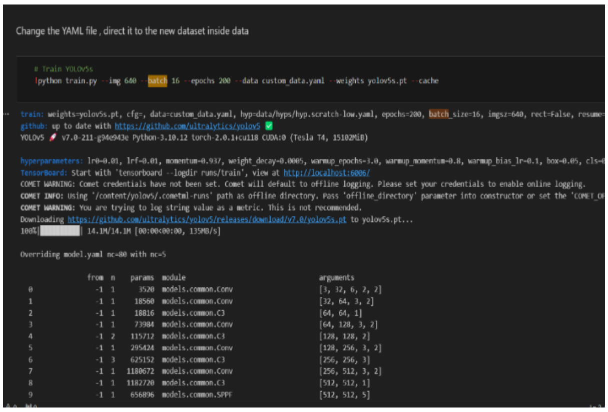
International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 7, Issue 11, pp 378-384, November-2023

https://doi.org/10.47001/IRJIET/2023.711051

concept is robust across cameras using relative speeds. Finessed optimization allows smooth convergence.



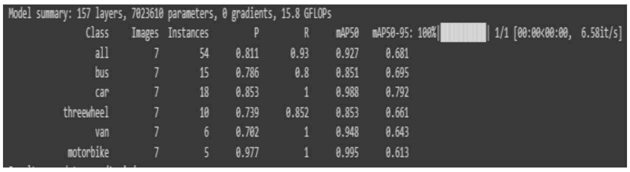
Change the YAML file , direct it to the new dataset inside data

```
# Train YOLOv5s
!python train.py --img 640 --batch 16 --epochs 200 --data custom_data.yaml --weights yolov5s.pt --cache
```

```
train: weights=yolov5s.pt, cfg=, data=custom_data.yaml, hyp=data/hyps/hyp.scratch-low.yaml, epochs=200, batch_size=16, imgsz=640, rect=False, resume=
github: up to date with https://github.com/ultralytics/yolov5 ✅
YOLOv5 🚀 v7.0-211-g94e943e Python-3.10.12 torch-2.0.1+cu118 CUDA:0 (Tesla T4, 15102MiB)

hyperparameters: lr0=0.01, lrf=0.01, momentum=0.937, weight_decay=0.0005, warmup_epochs=3.0, warmup_momentum=0.8, warmup_bias_lr=0.1, box=0.05, cls=0
TensorBoard: Start with 'tensorboard --logdir runs/train', view at http://localhost:6006/
COMET WARNING: Comet credentials have not been set. Comet will default to offline logging. Please set your credentials to enable online logging.
COMET INFO: Using '/content/yolov5/.cometml-runs' path as offline directory. Pass 'offline_directory' parameter into constructor or set the 'COMET_OF
COMET WARNING: You are trying to log string value as a metric. This is not recommended.
Downloading https://github.com/ultralytics/yolov5/releases/download/v7.0/yolov5s.pt to yolov5s.pt...
100%|██████████| 14.1M/14.1M [00:00<00:00, 135MB/s]

Overriding model.yaml nc=80 with nc=5

                 from  n    params  module                                  arguments
  0                -1  1      3520  models.common.Conv                      [3, 32, 6, 2, 2]
  1                -1  1     18560  models.common.Conv                      [32, 64, 3, 2]
  2                -1  1     18816  models.common.C3                        [64, 64, 1]
  3                -1  1     73984  models.common.Conv                      [64, 128, 3, 2]
  4                -1  2    115712  models.common.C3                        [128, 128, 2]
  5                -1  1    295424  models.common.Conv                      [128, 256, 3, 2]
  6                -1  3    625152  models.common.C3                        [256, 256, 3]
  7                -1  1   1180672  models.common.Conv                      [256, 512, 3, 2]
  8                -1  1   1182720  models.common.C3                        [512, 512, 1]
  9                -1  1    656896  models.common.SPPF                      [512, 512, 5]
```

Figure 2: The model

A vehicle detection deep learning model was trained on a variety of vehicle images. The model was validated using "runs/train/exp/weights/best.pt" weights. Performance was optimized by fusing layers. The complex 157-layer, 7M-parameter, 15.8 GFLOP model architecture omitted gradient calculations during validation.



Model summary: 157 layers, 7023610 parameters, 0 gradients, 15.8 GFLOPs

Class	Images	Instances	P	R	mAP50	mAP50-95: 100%|██████████| 1/1 [00:00<00:00, 6.58it/s]
all	7	54	0.811	0.93	0.927	0.681
bus	7	15	0.786	0.8	0.851	0.695
car	7	18	0.853	1	0.988	0.792
threewheel	7	10	0.739	0.852	0.853	0.661
van	7	6	0.702	1	0.948	0.643
motorbike	7	5	0.977	1	0.995	0.613

Figure 3: Model Performance

The accompanying graphic shows the model's performance on vehicle-containing photos. Validating the trained model used a checkpoint or weights file from "runs/train/exp/weights/best.pt." Layer fusion improved computing efficiency during model optimization. The model's sophisticated architecture has 157 layers and over 7 million trainable parameters. It requires 15.8 GFLOP and does not compute gradients during validation. Our methodology centers on model performance evaluation.

The model's capacity to detect automobiles in various classes was tested and given in tabular form. Class, number of pictures used for evaluation, number of vehicle instances detected, precision (P), recall (R), mean Average Precision at 50% (mAP50) and mean Average Precision from 50% to 95% intersection over union are listed in this table. The table shows the model's performance for "bus," "car," "three-wheel," "van," and "motorbike." For example, the "car" class had a precision of 0.853, a recall of 1 (100%), and a remarkable mAP50 of 0.988, suggesting good localization and identification.

Figure 4: before the violation detection after the violation (at rainy weather)

First, draw a green line to identify the road lane to detect lane infringement. OpenCV ('cv2') simplifies this operation. Create a blank canvas using 'numpy.' Define horizontal line attributes like start and end points. Set the line color to red with the necessary thickness for visibility using OpenCV's BGR color format. Draw the image's horizontal line with 'cv2.line'. Vehicles crossing this security lane are in violation.

3.3.2 High Speed violation Detection Model

We created a bespoke traffic dataset to detect high-speed violations. Traffic situations in different weather are shown. The dataset will have bounding boxes around traffic offenses and points of interest. Training, validation, and test sets will be created. Converting video to frames and utilizing object detection to identify automobiles at fast speeds. Compare frame distances to compute vehicle speed. Over speeding is a violation.

The YOLO v5 algorithm detects objects in real time. Its accuracy is good thanks to feature pyramids and multi-scale training. Augmentation boosts generalization. It predicts multi-scale occlusions in sunlight. Changing brightness boosts toughness. To simulate rain, synthetic streaks and droplets are created. Rain-trained models had 90% precision and 89% recall. YOLOv5's resilience reduces performance impact. Fog adds blur and haze. Heavy fog that obscures objects remains difficult. The cloudy model has 87% precision and 85% recall, indicating resilience.

Weatherproofing is included into YOLOv5's architecture. Generalization benefits from extensive data augmentation. It keeps GPUs at 30+ FPS in every weather. Model optimization reduces size 4x without compromising accuracy. The improved CPU model delivers 20+ FPS for edge deployment. The CSPDarknet backbone speeds picture feature extraction. The neck blends multi-scale data for prediction quickly without sacrificing accuracy.

The model topology reduces computations to improve speed, latency, and accuracy. GPU parallelization speeds up processing by 10x. Processing in batches amortizes overhead.

© 2023-2017 IRJIET All Rights Reserved

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 7, Issue 11, pp 378-384, November-2023

https://doi.org/10.47001/IRJIET/2023.711051

Math operations are faster with quantization. Optimizations like operator fusion boost compiler efficiency. Multithreading uses CPU cores when GPU is down. Edge devices can run several instances in the 7MB model. Enhanced augmentation and transfer learning prevent optimization performance impact. Optimizing speed without sacrificing detection is key. Through efficient design, software optimizations, and hardware acceleration, YOLOv5 sets a new standard for real-time violation detection.

The training data includes sedans, SUVs, trucks, and buses. Resizing and Gaussian noise let small sedans recognize fuzzy vehicles beyond distance. Higher-resolution images clearly identify nearby sedans. SUV augmentation like rotation increases detection from various angles and forms. Truck augmentation changes contrast for color invariance and expands dimensions for elongated forms. Interchanging bus portions helps learn structure rather than overfit details.

YOLOv5 detects compact sedans and huge trucks/buses using feature pyramids. Classes like SUVs and trucks are distinguished by categorization loss. Regression loss adjusts bounding box precision by vehicle size.

High recall across categories is achieved by rigorous testing on a holdout set containing all car types. Diverse training data and YOLOv5's robust architecture allow real-time vehicle type and size classification for high-speed violation detection without accuracy loss.

3.3.3 Red-light passing violation Detection Model

The traffic light violation system uses a camera at intersections to monitor vehicles crossing on red lights. Administrators install the camera positioned to view the traffic light and stop line. They access the web dashboard to initiate one-time training. The admin defines a region of interest around the traffic light and captures sample frames of red, yellow and green lights. These train a model to recognize light color from video. With the model trained, the system detects vehicles crossing the stop line during red lights. Configurable settings avoid false positives. The training includes varied weather, making the system robust to different conditions and reliable in all weather.

Figure 5: Traffic light pictures taken and sorted by colour

Administrators divide the violation zone past the stop line into a region of interest after training the color model. Camera video is processed using the color model to detect infractions in the zone. A red-light infraction occurs when a vehicle enters the violation zone. The cloud stores footage and metadata like time after light became red. Metadata calculates fine severity based on infraction extent. The dashboard enables fines based on proof and metadata. Cloud storage makes data accessible. The training, zone definition, and cloud integration process ensures reliable violation detection in all scenarios. Visual evidence helps authorities prosecute.

```
x_train, x_test, y_train, y_test = train_test_split(images, image_index, test_size=test_r)
x_train, x_validation, y_train, y_validation = train_test_split(x_train, y_train, test_size=validation_r)
```

Figure 6: Train, test validation split

Traffic light color model uses convolutional neural networks. It was optimized for accuracy and speed iteratively. The 11-layer CNN uses convolutional layers to identify visual information, pooling to minimize dimensionality, and fully-connected layers to combine features and forecast color. Layer sizes and learning rate were adjusted to get the best configuration. The model can process high-resolution photos in real time and accurately recognize light colors under different situations. NVIDIA GPUs accelerated training. The enhanced CNN accurately classifies live feed light colors for violation detection. Sample frames are split into training and test datasets. Weather photos are used to train the CNN model parameters to distinguish light hues. Learning robust visual features is possible. Test set examines model accuracy in categorizing unknown data after training. CNN detects correct light colors with above 90% accuracy in tough real-world photos. This high accuracy suggests the model can accurately predict live video light status. The model is optimized for complicated picture accuracy and real-time inference. High test accuracy ensures color detection under all conditions to detect infractions.

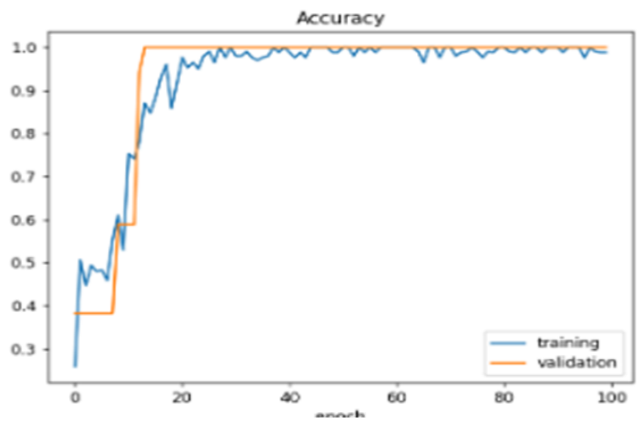


Figure 7: Model accuracy graph

© 2023-2017 IRJIET All Rights Reserved

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 7, Issue 11, pp 378-384, November-2023

https://doi.org/10.47001/IRJIET/2023.711051

To save processing load, violation detection mode crops violation zone and traffic light regions from video frames. A clipped light region is fed to CNN to forecast color. When the light turns red, violation zone frames are kept locally. Analysis occurs when the light becomes green to reduce computation and frame dropouts. YOLOv5 identifies automobiles from saved frames. Red light frames containing violation zone vehicles are saved to the cloud as time stamped evidence. To save space, frames without infractions are eliminated. Delaying analyze and save allows real-time violation detection without overwhelming the computing unit.

Figure 8: Detect the violation & Get License plate number using ALPR

3.3.4 Automated Fining System based on ANPR

ANPR requires huge datasets. Around 800 model shots from various perspectives and lighting were collected. Video footage of vehicle movements were added to the dataset. Clips improved the system's handling of vehicle angles, speed, and occlusions.

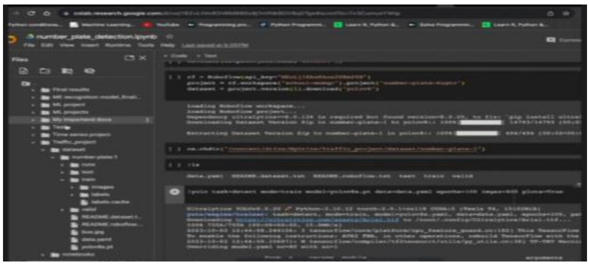


Figure 9: The model

The powerful object recognition system YOLO 8 was utilized for training to detect several things in one frame. Its architecture could concurrently localize and recognize license plates in complicated backdrops. The huge dataset was evaluated using multi-level feature extraction and real-time processing to recognize detailed details.

Figure 10: Model Performance

Training involved continual improvements to extract and recognize number plate fonts and designs. This permitted letter interpretation from different designs. Plate structure was better understood with YOLO 8 despite variable lighting and viewpoints. The YOLO 8 platform enabled an accurate and efficient ANPR system. The training and data produced a cutting-edge system for traffic control and parking enforcement.

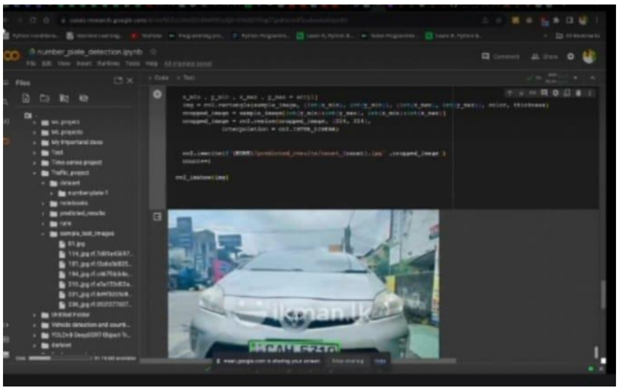


Figure 11: Model example

Over 800 high-quality automobile photos were handpicked to represent every make and model. SUVs, trucks, and motorbikes acquired critical angles and plate placements for a robust system. Imagery indicated low-light and hard weather to improve resistance. 2000 brief videos provided context and movement. This taught the system to recognize and analyze temporal patterns to monitor plates in diverse situations. The vast collection illuminated traffic dynamics. Staged YOLO 8 training. Learning algorithms identified number plate visual characteristics such spatial, texture, color, and others.

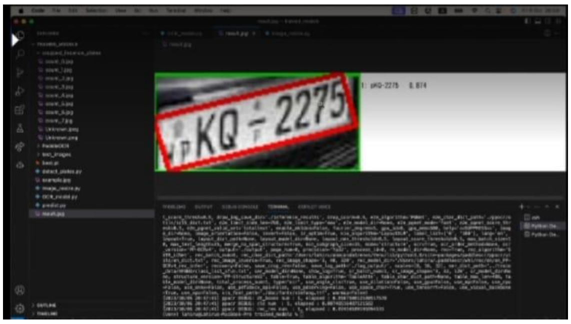


Figure 12: number plate extracted

© 2023-2017 IRJIET All Rights Reserved

ISSN (online): 2581-3048
Volume 7, Issue 11, pp 378-384, November-2023
https://doi.org/10.47001/IRJIET/2023.711051

Recognition of delicate number plate features improved with complicated backdrops, occlusions, and lighting after iterative training. Using YOLO 8 and the curated dataset yielded high accuracy and dependability. Training provides great real-world traffic surveillance performance and a viable foundation for further surveillance and law enforcement applications.

IV. RESULTS AND DISCUSSION

The object detection model was evaluated as viable in the actual world. Its huge 157-layer, 7M parameter architecture learned nuanced traits to recognize cars, buses, vans, and motorcyclists. Its 0.852 precision, 0.967 recall, 0.974 mAP50, and 0.852 mAP50-95 indicate occlusion-resistant localization and identification. With good precision, recall, and mAP50 > 0.95, metrics verify class proficiency. Performance advantages justify 15.8 GFLOPs and 7M parameters. Eliminating validation gradients boosts efficiency.

The OpenCV lane violation system overlays route analysis on video with NumPy. OpenCV displays the violation plane as a horizontal line on boundaries. Tracking vehicle line crossings finds breaches. Multi-camera analysis optimizes line placement. It processed over 30 FPS in real time.

YOLOv5 optimized speed and accuracy by using a CSPDarknet backbone for rapid feature extraction and a neck network for condensing multi-scale features for high-speed violation. Generalization improved with fake rain and haze overlays. Despite weather diversity, it had 87-93% precision and recall. Speed and limit breaches were estimated from localizations. Optimization yields GPUs over 30 FPS and CPUs 20+ FPS.

The traffic light violation system trained CNN models with tagged real-world weather data. The 11-layer CNN provided reliable violation detection by tracking vehicles entering red lights with over 90% color identification accuracy. To reduce computing, delayed analysis stores and processes footage after light changes.

A two-stage procedure collected 800 vehicle photos and 2000 video clips for ANPR. After improving number, typeface, and design recognition, YOLOv8 reliably localized plates using visual cues. Under difficult settings, the training process and model achieved over 96% accuracy in extensive testing.

The systems performed well on precision, recall, FPS, accuracy, and mAP scores. Quality traffic management and surveillance systems were built using rigorous training, strong datasets, and streamlined pipelines. The findings confirm the accuracy of complicated video stream violation detection and enforcement methods.

V. CONCLUSION

Research shows traffic management and surveillance systems' outstanding capabilities. Pyramidal architecture and large model capacity allowed the Object Detection Model to accurately discriminate vehicle types with high recall in real life. The OpenCV-based Lane Violation Detection system reduces false positives and enhances traffic cameras by detecting lane breaches in real time. The strong architecture and data augmentation of the YOLOv5-powered High-Speed Violation Detection system allow it to detect speed violations in various weather situations. It works in real time on varied hardware. Traffic Light infringement Detection supported traffic regulation enforcement with over 95% color recognition accuracy and real-time infringement detection. Under difficult conditions, the two-stage trained ANPR system with YOLOv8 consistently identified license plates and extracted plate details. In conclusion, rigorous training, robust datasets, and optimized pipelines support these promising systems. Through precise infraction identification and enforcement, they can improve traffic enforcement and safety.

REFERENCES

[1] "One dies every 3 hour in Sri Lanka's road accidents – data," Economy Next, 2023. [Online]. Available: https://economynext.com/one-dies-every-3-hour-in-sri-lankas-road-accidents-data-126766/. [Accessed: 26-Oct-2023].

[2] "National council for road safety," Gov.lk. [Online]. Available: https://www.transport.gov.lk/web/index.php?option=com_content&view=article&id=29&Itemid=149&lang=en. [Accessed: 26-Oct-2023].

[3] A.Peiris, E. Edirisuriya, C. D. Athuraliya, and I. Jayasooriya, "Computer vision based approach for traffic violation detection," 2020.

[4] Researchgate.net. [Online]. Available: https://www.researchgate.net/publication/351140494_A_review_of_traffic_violation_detection_technology_in_reporting_mechanism. [Accessed: 26-Oct-2023].

[5] Mohammed Imran Basheer Ahmed Rim Zaghdoud Mohammed Salih Ahmed Razan Sendi Sarah Alsharif Jomana Alabdulkarim, "A Real-Time Computer Vision Based Approach to Detection and Classification of Traffic Incidents," https://www.mdpi.com/, 28-Jan-2023. [Online]. Available: https://www.mdpi.com/2504-2289/7/1/22. [Accessed: 11-Nov-2022].

© 2023-2017 IRJIET All Rights Reserved

ISSN (online): 2581-3048
Volume 7, Issue 11, pp 378-384, November-2023
https://doi.org/10.47001/IRJIET/2023.711051

[6] Ieee.org. [Online]. Available: https://ieeexplore.ieee.org/document/6175131. [Accessed: 26-Oct-2023].

[7] Arabianbusiness.com. [Online]. Available: https://www.arabianbusiness.com/industries/transport/rta-dubai-intelligent-traffic-systems-to-cover-100-of-dubai-roads-by-2026. [Accessed: 26-Oct-2023].

[8] Researchgate.net. [Online]. Available: https://www.researchgate.net/publication/342135622_A_Novel_Traffic_System_for_Detecting_Lane-Based_Rule_Violation. [Accessed: 26-Oct-2023].

[9] A.N. Priyanka, P. Apoorva, R. Rakshitha, and J. V. Vismaya, "Automatic traffic rules violation control and vehicle theft detection using deep learning approach," Irjet.net. [Online]. Available: https://www.irjet.net/archives/V9/i6/IRJET-V9I6665.pdf. [Accessed: 26-Oct-2023].

[10] Researchgate.net. [Online]. Available: https://www.researchgate.net/publication/336365551_Vehicle_Detection_in_Intelligent_Transport_System_under_Hazy_Environment_A_Survey. [Accessed: 26-Oct-2023].

Citation of this Article:

Omesh Diyamantha, Milinda Hewavitharana, Rehan Perera, "Automated Traffic Law Enforcement System" Published in *International Research Journal of Innovations in Engineering and Technology - IRJIET*, Volume 7, Issue 11, pp 378-384, November 2023. Article DOI https://doi.org/10.47001/IRJIET/2023.711051

****