

A Transfer Learning Approach for Identification of Two-Wheeler Brand and Model

S Mohammed Mansoor Roomi, Lokesh V S*, Shankar Mahadevan G, Priya K

Thiagarajar College of Engineering,
Electronics and Communication Engineering,
India

smmroomi@tce.edu, {vslokesh10, shankarmahadeva12901, priya5586}@gmail.com

Abstract. Motorcycles were originally designed to provide safer, more efficient, and more comfortable rides, but they are now also used for criminal purposes. Due to overspeeding, motorcycles often get involved in accidents, and it is challenging for law enforcement officials to identify the culprits from CCTV footage or spectator accounts. This paper presents a solution to this challenge by using a pre-trained deep learning model to detect, classify, and identify motorcycle models. To overcome the limited availability of annotated bike databases, this proposed work created a new bike dataset that includes 5000 annotated images sourced from major search engines, CCTV footage, and manual captures from bike showrooms. Then the bike brand was identified in 27 classes by the Faster RCNN pre-trained model and achieved an accuracy of 94.35%. The proposed model was compared with the other pre-trained models such as YOLO V5 and MobileDET, among these, the Faster RCNN provided better identification accuracy.

Keywords. Bike dataset, wheel rim, headlamp, yolo V5, MobileDET, faster RCNN, imbalanced learning, and weighted loss learning.

1 Introduction

Two-wheelers are a common form of transportation in India but are also frequently involved in serious accidents and crimes. As a result, research on smart traffic systems, including vehicle detection, identification, enumeration, and traffic statistic estimate, has become increasingly popular. Researching traffic estimates, speed calculations, motorcycle helmet use, vehicle tracking, as well as occlusion analysis requires segmenting automobiles on public roadways.

The proposed work aims to address the challenge of detecting and classifying motorbikes on public roads. As our world moves faster, the number of vehicles on the road is increasing daily, along with a corresponding rise in crimes and accidents involving bikes. This increase appears to be linear.

To find who is the cause of the accident or the culprit in a crime. The investigation officer enquires about the people in and around the place where the incident has taken place. but they won't have recognized or seen the person's face or the number plate they may have seen only what color the bike was and how did the bike look like or the color of the bike.

Hence to find a solution to this problem, the proposed work is to find the bike model using the bike's wheel rim and bike headlamp. we could have used the number plate to identify the bike and trace out the person who rode the bike but sometimes the number plate may be damaged or sometimes the numberplate may be slightly faded and which is not visible at all or sometimes the bike maybe without number plate also hence without wheel rim a bike cannot move or is not complete hence with the help of wheel rim and headlamp we are going to detect the bike model. Figure 1 shows a sample image of the bike with localization of the wheel rim and headlamp.

2 Related Works

Research into computer vision and image processing (IP) techniques for automating vehicle classification has been ongoing.



Fig. 1. Sample images from the collected data set

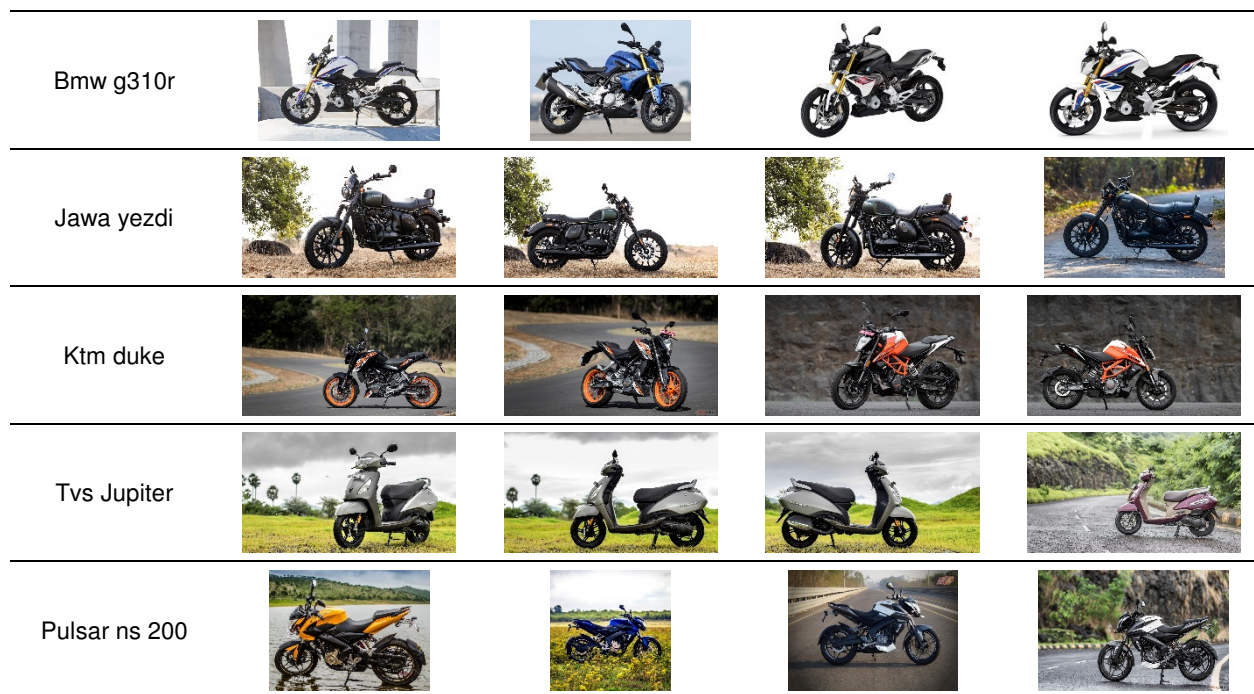


Fig. 2. Examples of annotated bike categories are presented, demonstrating variations in both form and pose

One possible approach involves pre-processing images using IP techniques to enhance their suitability for input into a machine-learning model. For example, in [1], a stationary camera was used to record footage of vehicles, which were then classified using three levels of processing and

appropriate camera calibration. Similarly, [2] suggested using IP techniques, Invariant Feature Transform (IFT), K-means clustering, and euclidean distance matching to classify vehicles into different categories for Electronic Tolling Collection.

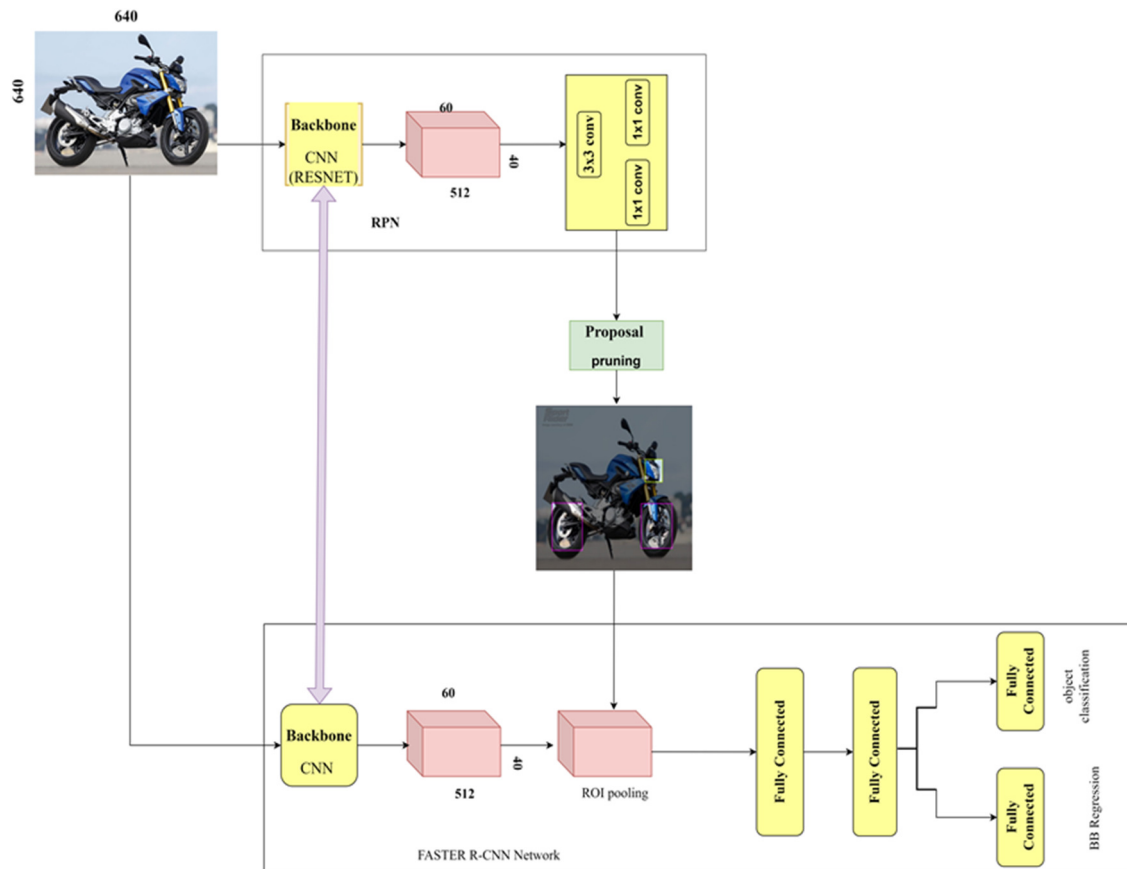


Fig. 4. Flow of faster RCNN architecture

Another example is a pre-crash detection system, as described in [3], which utilized a front-facing low-light camera to detect vehicles and avoid accidents. This system used a combination of multiscale-driven hypothesis generation and appearance-based hypothesis verification to achieve good results in all kinds of weather and road conditions.

In [4], Sonoda et al. proposed a system for detecting moving objects, such as vehicles and pedestrians, at an intersection, which warns drivers of potential dangers. The system employed a mixture of Gaussians for object detection and the Lucas-Kanade Tracker (LKT) algorithm for pedestrian tracking.

In [5], researchers used a camera to capture images of vehicles against a static background, allowing them to calculate vehicle length and width and classify the vehicle as either a Light Motor

Vehicle (LMV) or a Heavy Motor Vehicle (HMV). The goal of this research was to determine a suitable toll rate.

In [6], an automated vehicle classification system was proposed to distinguish between motorcycles and sedans, using commercially available light curtains to produce images and two machine learning methods, K Nearest Neighbor (KNN) and decision tree learning, to classify the vehicles. Finally, [7] proposed a computer vision system to detect and segment motorcycles that were partially occluded by other vehicles, using a helmet detection system to identify motorcycles.

However, this approach requires significant information, such as helmet radius, camera angle, and camera height, to function effectively [8] developed a system that can identify vehicles even if they are partially occluded and can continue

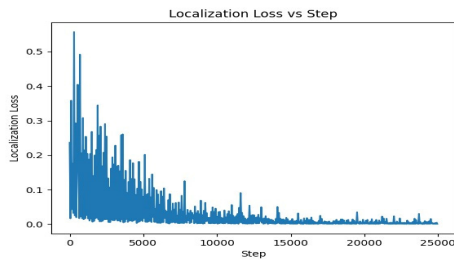


Fig. 5. Plot of localization loss vs step

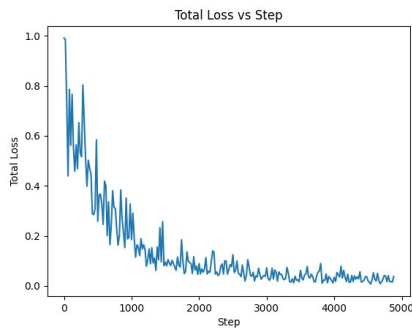


Fig. 6. Plot of total loss vs step

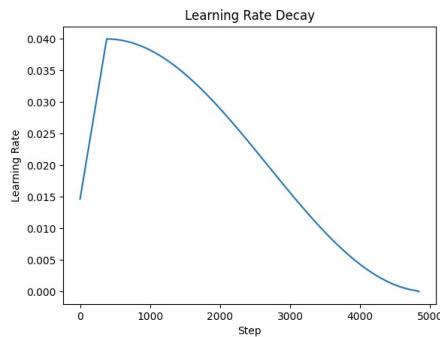


Fig. 7. Plot of learning rate vs step

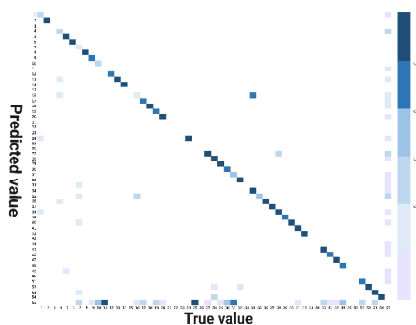


Fig. 8. Confusion matrix of bike model classification

tracking the vehicle despite occlusions. In [9], a computer vision system was developed for detecting bicycles, pedestrians, and motorcycles.

This system used Gabor filtering to detect motion and a Histogram of Oriented Gradient (HOG) descriptors to classify two-wheeled vehicles and pedestrians. Chiverton et al. [10] proposed a system for automatically classifying and tracking motorcycle riders with and without helmets, using histograms derived from the head region of riders to train an SVM classifier.

Although this system achieved high accuracy, the number of test images was limited. The current research work mainly focused on vehicle classification with limited classes and quantity. But there is a need for the bike model dataset with large classes for automation of the bike model identification.

This proposed work created a bike model dataset and pre-trained the Faster Region Convolution Neural Network (RCNN) model for the identification of the bike model. The major contributions of the proposed method are:

- Creation of a bike dataset with 27 classes.
- Proposal of a faster RCNN deep learning model for bike model identification.
- Comparison of proposed work against the existing pre-trained model.

3 Proposed Methodology

The proposed methodology comprises four stages: database creation, augmentation, and training and testing phases to identify motorcycle models using categorical images. The dataset was created by collecting images from various sources LIKE search engines, movies, CCTV footage, and bike showrooms, resulting in a diverse collection of two wheelers.

Data cleaning was performed to remove unwanted images, resulting in a dataset of 1,500 motorcycle images from 27 categories, which were augmented to obtain a total of 5,000 distinct images stored in the database.

However, the target variable showed an imbalance in the number of instances across different classes. Sample images from the created bike dataset are presented in Figure 2. The Faster

Table 1. Labels of bike models

SI.No	Label	Sample Count
1	BMW g310 r	1,863
2	Ducati panigale	281
3	Gixxer SF	214
4	Hero pleasure+	137
5	Hero splendor +	136
6	Honda Activa	280
7	Honda shine	152
8	Honda Unicorn	260
9	Jawa 42	175
10	Jawa yezdi	53
11	Kawasaki ninja	149
12	Ktm adventure	207
13	Ktm duke	239
14	Ktm RC	57
15	Pulsar ns 200	484
16	Pulsar rs	477
17	Super splendor	132
18	Suzuki Access	287
19	Tork kratos	264
20	Tvs apache rr	468
21	Tvs apache rtr	167
22	Tvs Jupiter	210
23	Tvs ntorq	202
24	Tvs sport	171
25	Yamaha fzs	168
26	Yamaha mt	132
27	Yamaha r15	137

RCNN is a widely used object detection framework that was introduced by Ross Girshick in 2015.

This architecture employs Convolutional Neural Networks (CNNs) for object detection, similar to other popular designs such as You Look Only Once (YOLO) and Single Shot Detector (SSD).

When given an image and bounding boxes as input, the Faster R-CNN network processes the entire image using multiple convolutional and max pooling layers to generate a convolutional feature map.

In this study, the Faster RCNN was used to detect and identify objects in the created bike dataset. To generate feature maps of sizes 60, 40, and 512, the image is passed through the ResNet backbone of the CNN.

For proposal generation, the Region Proposal Network (RPN) is utilized, which benefits from

weight sharing between the Faster R-CNN detector backbone and the RPN backbone.

The input image is resized to 640×640 and is fed into the CNN's backbone to initiate the RPN. This study introduces backbone networks like ResNet, which have a network stride of 16, leading to significantly smaller output features compared to the input image. Each point on the output feature map corresponds to two points 16 pixels apart in the input image.

The proposed deep detector network aims to identify whether there is an object present in the input image at each point in the output feature map and estimate its size.

To achieve this, a set of "Anchors" is placed on the input image for each position on the output feature map generated by the backbone network. These anchors point to potential objects of various sizes and aspect ratios that could be present.

The network then determines which of the k -associated anchors in the input image contain objects and adjusts their coordinates to provide bounding boxes as "Object proposals" or areas of interest.

To create 512-d feature maps for each location, the backbone feature map undergoes a 3×3 convolution with 512 units.

Two sibling layers are added: a 1×1 convolution layer for object classification with 18 units and a 1×1 convolution layer for bounding box regression with 36 units. The classification branch's 18 units output a probability of whether an object is present at each location in the backbone feature map.

The regression branch's 36 units produce the size of the object proposals for each point in the backbone feature map. The regression coefficients of each of the 9 anchors for each point in the backbone feature map are obtained using this output.

The coordinates of the anchors containing objects are then adjusted using these regression coefficients

The backbone feature map features are obtained using the ROI pooling layer, which pools them based on RPN-proposed bounding box boundaries. The ROI pooling layer selects the area that matches the proposal from the backbone feature map, divides it into a fixed number of sub-windows, and performs max-pooling over these sub-windows to produce an output of a fixed size.

Table 2. Hyper tuning parameters of the faster RCNN model

Parameter	Value
Optimizer	Adam
Mini Batch Size	4
Device Type	GPU
Learning Rate	0.04
l2Regularization	0.013333
Epoch	2000

Table 3: Faster RCNN accuracy

Average Precision (AP)	0.9435
Average Recall (AR)	0.738
Localization loss	0.065122
Classification loss	0.167798
Regularization loss	0.000000
Total loss:	0.275017

The size of the ROI pooling layer's output is $(N, 7, 7, 512)$, where N is the number of proposals generated by the area proposal algorithm.

These features are passed through two fully connected layers before being fed into the sibling classification and regression branches. Two objects need to be labeled in a bike model detection task, and the labels for each object can be either 'headlamp' or 'wheel rim'.

The weights assigned to each label are $W_1 = 0.6$ for 'headlamp' and $W_2 = 0.4$ for 'wheel rim'. We assign a higher weight to the headlamp (0.6) than to the wheel rim (0.4) as the headlamp is more significant than the wheel rim. There are two possible label conditions:

- Both objects have the same label.
- Both objects have different labels.

To determine the label for an object, we use the following process:

- If both objects have the same bike model label, then the label for the object is the same as the label that both objects share.
- If one object has one bike model label and the other has a different bike model label, the proposed work uses the weighted formula $W_1 S_1 + W_2 S_2$ to determine the label for the object. The formula considers the confidence score for each label and the weight assigned to each

label. The confidence scores are calculated using a machine learning model, and they reflect the probability of each label is correct.

Then compare the values of $W_1 S_1$ and $W_2 S_2$ to determine which label is more appropriate for the object. If $W_1 S_1$ is greater than $W_2 S_2$, then the object is labeled as the label described by the headlamp, and if $W_2 S_2$ is greater than $W_1 S_1$, then the object is labeled as the label described by the wheel rim. $W_1 S_1 \geq W_2 S_2$.

4 Results and Discussions

The results and discussion of the proposed approach are described in this section. The collection of bike images that were developed was divided into an 80:20 ratio. 80 percent of the samples were chosen for training, 10 percent for validation, and the rest for testing.

The proposed algorithm was trained using Google Colab, which was set up on an HP server with 8GB of RAM. The training images were reduced in size at the pre-processing stage to 640×640 . The Faster RCNN pre-trained model was then trained using the resized image. Table 2 contains the default settings for the training parameters of the proposed model.

To increase accuracy, these parameters are modified, including the optimizer, learning rate, and epochs. Figure 3 depicts the proposed pre-trained model's training plot. This model's training recognition accuracy improves to 94.35% once these parameters are adjusted. Table 3 represents the performance metrics of a FASTER RCNN that is designed to detect and classify bikes using the wheel rim and headlamp of a motorcycle.

Average Precision (AP), measures the accuracy of the model in terms of its ability to correctly identify the presence of a bike, as well as its ability to correctly identify the specific type or model of the bike.

The value of 0.9435 indicates a relatively high level of accuracy. Average Recall (AR), indicates the proportion of actual bike instances that were correctly identified by the model.

The value of 0.738 suggests that while the model is accurate, there is room for improvement in terms of correctly identifying all instances of

Table 4. Labels in the confusion matrix

1	super splendor -headlamp	28	hero pleasure - wheel
2	super splendor wheel	29	Ducati Panigale v4
3	fzs headlamp	30	Panigale headlamp
4	fzs wheel	31	access headlamp
5	Jawa 42 headlamp	32	access wheel
6	Jawa 42 wheel	33	honda shine headlamp
7	gixxer sf headlamp	34	honda shine wheel
8	gixxer sf wheel	35	apache rtr headlamp
9	tork Kratos headlamp	36	apache rtr wheel
10	tork Kratos	37	ktm duke headlamp
11	r15 headlamp	38	ktm duke wheel
12	r15 wheel	39	tvS sport headlamp
13	ktm adventure headlamp	40	tvS sport wheel
14	ktm adventure wheel	41	honda activa headlamp
15	mt15 headlamp	42	honda activa wheel
16	mt15 wheel	43	jawa yezdi headlamp
17	apache rr headlamp	44	jawa yezdi wheel
18	apache rr wheel	45	honda unicorn headlamp
19	bmw G310R headlamp	46	honda unicorn wheel
20	bmw G310R wheel	47	pulsar rs 200 headlamp
21	ktm rc headlamp	48	pulsar rs 200
22	ktm rc wheel	49	pulsar ns 200 headlamp
23	hero splendor plus-headlamp	50	pulsar ns 200
24	hero splendor plus-wheel	51	Jupiter headlamp
25	kawasaki ninja headlamp	52	Jupiter wheel
26	kawasaki ninja wheel	53	ntorq headlamp
27	hero pleasure - headlamp	54	ntorq wheel

bikes in the dataset. Localization loss measures the error in the model's ability to accurately localize the object in the image. The value of 0.065122 indicates that the model is performing well in this regard.

The classification loss measures the error in the model's ability to accurately classify the type or model of the bike. The value of 0.167798 suggests that the model is performing reasonably well in this regard, but there is room for improvement. Regularization loss measures the degree of regularization in the model, which is designed to prevent overfitting.

The value of 0.000000 suggests that the model does not overfit the training data. Total_loss is the sum of the three previous loss values and represents the overall error of the model.

The value of 0.275017 indicates that the model is performing well, but there is still room for improvement in terms of correctly identifying all

instances of bikes in the dataset, as well as accurately classifying the type or model of the bike.

Figure 5 shows a decreasing trend in the loss as the number of steps increases, with some fluctuations due to noise in the data or variations in the optimization process.

At the beginning of training, the localization loss is high and the model's predictions are poor, as indicated by the high point at the left side of the graph. However, as the training progresses, the loss decreases steadily, with occasional fluctuations, until it reaches a minimum.

Figure 6 shows the total loss decreases over the course of training, as the model becomes better at making predictions.

The graph shows some fluctuations in the loss due to noise in the data or variations in the optimization process.

At the beginning of training, the loss is high because the model's predictions are poor.

Table 5. Performance analysis of proposed model with other models

Deep Learning Techniques	Accuracy
Mobile Det	89.5%
Yolo V5	90%
Proposed Network (Faster Rcn)	94.35%
Mobile Det	89.5%

However, as the training progresses, the loss decreases steadily, with occasional fluctuations, until it reaches a minimum.

In Figure 7, the learning rate schedule for the training process is illustrated. Initially, a high learning rate, such as 0.04, is set to allow the optimization algorithm to quickly search the parameter space and find a good initial solution.

During the course of training, the learning rate gradually decreased based on a pre-defined schedule. In the given example, the learning rate is reduced by a factor of 10 after every 1000 steps.

Therefore, after 2000 steps, the learning rate is reduced to 0.03, and after 3000 steps, it is further reduced to 0.02, and so on. The proposed work's performance can be evaluated using the confusion matrix, as shown in figure 8 And the labels of each model are shown in table 4.

Precision, recall, and the F1 score are used to evaluate the proposed model performance. There are 27 classes, each class has two labels hence 54 subclasses in these 54 subclasses in 54 subclasses 48 subclasses have accuracy above 90%. The proposed model outperforms the others since 94.35% of the Bike models were correctly classified in the testing phase.

Table 5 shows the accuracy of each model on the task, as a percentage. MobileDET achieved an accuracy of 89.5%, YOLO v5 achieved an accuracy of 90%, and the proposed network using Faster R-CNN achieved the highest accuracy of 94.35%.

5 Conclusion

Detecting and classifying motorbikes on public roads is a challenging task, but it has become increasingly important due to the rising number of

bike-related crimes and accidents. The proposed work focuses on identifying the bike model using the bike's wheel rim and headlamp, as these components are often visible even if the number plate is damaged or not present.

The proposed work consists of bike model database creation, training the bike model by Fater RCNN, and testing the bike model identification.

This work achieved better identification accuracy of 94.35%. The proposed Fater RCNN model compared with YOLO v5 and MobileDET. Ultimately, this work has the potential to make our roads safer and help law enforcement agencies in their efforts to prevent and solve crimes.

References

1. **Ozkok, F. O. (2017).** A new approach to determine Eps parameter of DBSCAN algorithm. *International Journal of Intelligent Systems and Applications in Engineering*, Vol. 4, No. 5, pp. 247–251. DOI: 10.18201/ijisae.2017533899.
2. **Ng, J. Y., Tay, Y. H. (2012).** Image-based vehicle classification system. *Proceedings of the 11th Asian-Pacific ITS Forum and Exhibition*, pp. 1–11. DOI: 10.48550/ARXIV.1204.2114.
3. **Zehang, S., Miller, R., Bebis, G., Dimeo, D. (2002).** A real-time precrash vehicle detection system. *Proceedings of the 6th IEEE Workshop on Applications of Computer Vision*, pp. 171–176. DOI: 10.1109/acv.2002.1182177.
4. **Tan, J. K., Ishikawa, S., Sonoda, S., Miyoshi, M., Morie, T. (1970).** Moving objects segmentation at a traffic junction from vehicular vision. *ECTI Transactions on Computer and Information Technology*, Vol. 5, No. 2, pp. 73–88. DOI: 10.37936/ecti-cit.201152.54239.
5. **Shobha-Rani, B. R., Suparna, B. M., Teja, K. S. (2015).** Classification of vehicles using image processing techniques. *International Journal of Engineering Research and Technology*, Vol. 3, No. 21, pp. 1–4.
6. **Sarikan, S. S., Ozbayoglu, A. M., Zilci, O. (2017).** Automated vehicle classification with

- image processing and computational intelligence. *Procedia Computer Science*, Vol. 114, pp. 515–522. DOI: 10.1016/j.procs.2017.09.022.
7. **Giron, N. N. F., Billones, R. K., Fillone, A., Del-Rosario, J. R., Cabatuan, M., Bandala, A., Dadios, E. P. (2020).** Motorcycle rider helmet detection for riding safety and compliance using convolutional neural networks. *IEEE 12th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management*, pp. 1–6. DOI: 10.1109/hnicem51456.2020.940 0149.
 8. **Yuan, Y., Zhang, J., Wang, Q. (2018).** Bike-person re-identification: A benchmark and a comprehensive evaluation. *IEEE Access*, Vol. 6, pp. 56059–56068. DOI: 10.1109/access.2018.2872804.
 9. **Sun, Z., Bebis, G., Miller, R. (2002).** On-road vehicle detection using Gabor filters and support vector machines. *Proceedings of the 14th International Conference on Digital Signal Processing*, Vol. 2, pp. 1019–1022. DOI: 10.1109/ICDSP.2002.1028263.
 10. **Chiverton, J. (2012).** Helmet presence classification with motorcycle detection and tracking. *IET Intelligent Transport Systems*, Vol. 6, No. 3, pp. 259. DOI: 10.1049/iet-its.2011.0138.
 11. **Tabassum, S., Ullah, S., Al-Nur, N. H., Shatabda, S. (2020).** Poribohon-BD: Bangladeshi local vehicle image dataset with annotation for classification. *Data in Brief*, Vol. 33, pp. 106465. DOI: 10.1016/j.dib.2020.106465.
 12. **Espinosa, J., Velastin, S., Branch, J. (2018).** Motorcycle detection and classification in urban scenarios using a model based on faster r-CNN. *Proceedings of the 9th International Conference on Pattern Recognition Systems*, pp. 91–96 DOI: 10.1049/cp.2018.1292.
 13. **Regenwetter, L., Curry, B., Ahmed, F. (2021).** Biked: a dataset for computational bicycle design with machine learning benchmarks. *Journal of Mechanical Design*, Vol. 144, No. 3, pp. 1–19. DOI: 10.1115/1.4052585.
 14. **Figueiredo, A., Brayan, J., Reis, R. O., Prates, R., Schwartz, W. R. (2021).** MoRe: A large-scale motorcycle re-identification dataset. *IEEE Winter Conference on Applications of Computer Vision*, pp. 4033–4042. DOI: 10.1109/wacv48630.2021.00408.
 15. **Wang, H., Hu, Z., Guo, Y., Yang, Z., Zhou, F., Xu, P. (2020).** A real-time safety helmet wearing detection approach based on CSYOLOv3. *Applied Sciences*, Vol. 10, No. 19, pp. 6732. DOI: 10.3390/app10196732.
 16. **Kocamaz, M. K., Gong, J., Pires, B. R. (2016).** Vision-based counting of pedestrians and cyclists. *IEEE Winter Conference on Applications of Computer Vision*, pp. 1–8. DOI: 10.1109/wacv.2016.7477685.

Article received on 09/03/2023; accepted on 30/03/2023.

**Corresponding author is Lokesh V S.*