Vehicle Detection & Classification Using Integration of YOLOv3 & ORB Algorithm: A Deep Learning Approach

Shihabudeen H[1, *], Dr. Rajeesh J[2] and Dr. Umesh P[3]
[1]*College of Engineering Kidangoor, APJ Abdul Kalam Technological University, Kerala, India*
[2]*College of Engineering Perumon, Kerala, India*
[3]*College of Engineering Thalassery, Kerala, India*
[1]*shihabudeenh@ce-kgr.org* , [2]*rajeesh26071669@gmail.com*, [3]*toumesh@gmail.com*

Abstract

In deep learning techniques, Image processing and Computer vision ways of approach are becoming much more popular and are also widely used as it gives much better results compared to any other traditional methods. Vehicle identification and categorization are becoming more significant in intelligent transportation systems and visual traffic surveillance systems in recent years. Several recent studies show that the results can be more accurate if they had checked data imbalances. Also, some of the studies did not use proper extraction methods, which are crucial for the classification stage. So, to overcome these barriers, this paper proposes an effective deep learning-based technique where the following are the main stages: a) Data collection which is captured from surveillance cameras IVS-1 and IVS-2 b) Vehicle detection using Yolov3 c) Feature extraction using ORB algorithm d) Vehicle classification for classifying certain types of vehicles from the image instances. The evaluation of the proposed system (YOLO-ORB) with other state-of-art models depict a greater performance on various measures (accuracy:0.97, sensitivity:0.96, specificity:0.96, precision:0.95).

Keywords: *Classification, Deep learning, fusion, Infrared Image, Vehicle Detection, Visible Image, Yolov3.*

1. Introduction

The growth of vehicle traffic has resulted in issues such as traffic accidents, congestion, and air pollution. Traffic accidents are one of the most difficult issues to address. It's critical to have automated techniques for checking the suspected automobiles while conducting criminal investigations related to traffic accidents. To address this, numerous nations have developed Intelligent Transportation Systems (ITS) [1-3]. These systems recognize and categories cars using data from video and infrared cameras, as well as sound and vibration sensors. Traditional vehicle detection and categorization technologies, on the other hand, are prohibitively expensive. Furthermore, since these systems require a considerable quantity of hardware, they are difficult to be installed and run. Nonetheless, traffic surveillance cameras have been continuously put on key highways over the previous decade to monitor traffic. Making efficient use of these cameras for data collecting is crucial [4].

Moving vehicle detection and categorization are required for real-time smart surveillance systems in ITS to work. To detect moving vehicles in video frames,

foreground extraction is employed. The inter-frame difference technique [5], the background extraction method [6], and the optical flow estimation method [7] are all approaches for recognizing vehicles. The broad study of vehicle identification and classification using deep learning is depicted in Figure 1.

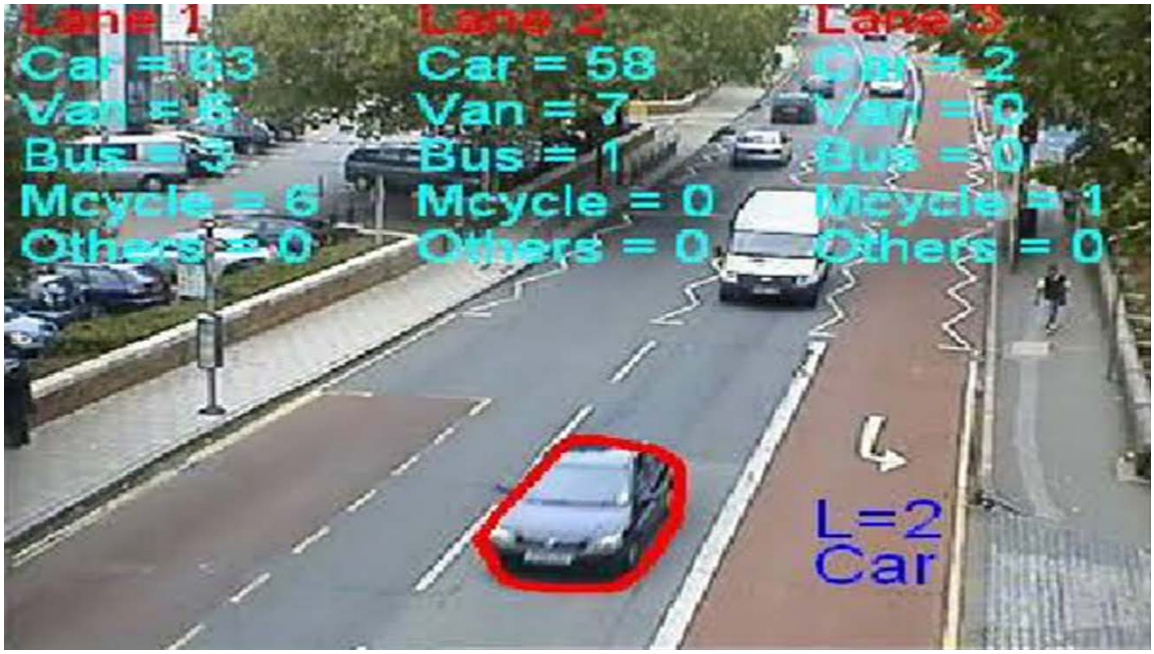

Figure 1. Vehicle detection and classification using deep learning: a deep learning perspective instance of an analytical image.

1.1 Key Highlights

This paper brings an effective deep learning perspective of vehicle detection and classification. Following are the main objectives discussed in this paper.

- How vehicle identification and categorization can be achieved using an advanced deep learning method.
- The Yolov3 network is in charge of vehicle detection and categorization.
- Feature extraction of each image is obtained using the ORB algorithm.
- Since using a large dataset over the advanced capability of the YOLOv3 network, the class imbalance is mitigated and thereby a greater accuracy is attained.
- When compared to state-of-the-art models, the recommended model surpasses all others.

Organization of the paper: The literature review is given in Section 2, the methodology is illustrated in Section 3, the performance analysis is explained in Section 4 and Section 5 has the conclusion.

2. Literature Recview

Yelmaz et al. [7] used R-CNN and Faster R-CNN deep learning algorithms to effectively train their vehicle detector on a sample vehicle data set, and then tested it on test data to increase the trained detector's success rate by giving efficient vehicle identification results. The working approach was broken down into six essential steps: The data set was loaded, the convolutional neural network was built, and the training settings

were specified. Faster R-CNN object detector training and trained detector assessment were the steps involved. Faster R-CNN and R-CNN deep learning approaches were also included in the study and experimental analytic comparisons with vehicle identification outcomes were noted.

Wang et al. [8] devised an innovative two-stage method for detecting automobiles and recognizing brake lights in real-time from a single picture. Rather than extracting pair taillights explicitly as in earlier techniques, they used the vehicle rear appearance picture. A multi-layer sensory neural network was used to learn "Brake Lights Patterns" (BLP) from a big database. The deep classifier could classify automobiles as "brake" or "normal" based on an image. Using a camera and multi-layer lidar, the automobile could be recognized rapidly and precisely (IBEO Lux fusion system). To increase detection speed and robustness, researchers used road segmentation and a novel vanishing point area of interest determination approach. The suggested approach's robustness and efficiency were demonstrated by experimental findings obtained on various genuine on-road films.

Major et al. [9] showed a deep-learning-based vehicle recognition system that employs an image-like tensor instead of the peak-detection-generated point cloud. To decrease false-positive rates, Shin et al. [10] presented a real-time vehicle identification technique based on deep learning. They also tested the algorithm on an embedded device to ensure that it worked in real-time. According to experimental data, when the deep learning model is utilized in the vehicle validation stage, the accuracy rate improves. The vehicle detecting modules process at a rate of roughly 15 frames per second on average.

For vehicle recognition, Espinosa et al. [11] examined two deep learning models. An urban video series was investigated using Alex Net and Faster R-CNN. A variety of investigations were conducted to establish the accuracy of detection, the rate of failure, as well as the time necessary to complete the assignment. The findings allow for important inferences regarding the designs and strategies used to build a video detection network. This paved the way for more study in this sector.

Hsu et al. [12] proposed a fast region-based convolutional neural network for vehicle recognition that is, a simplified (R-CNN). Fast R-CNN, a deep convolutional network-based approach for object recognition, is well-known. To detect vehicles, Shi et al. [13] established a targeted search strategy and a target identification model based on Fast R-CNN. By preprocessing the sample image and creating a new network architecture, the technique optimizes the model. The experiment uses the publicly available KITTI dataset for training validation and the self-collected BUU-T2Y dataset for testing to optimize the model using preprocessing. As part of the incremental learning process, the KITTI dataset is integrated with the BUU-T2Y dataset, based on the original data set.

Nguyen et al. [14] offer a better framework for quick vehicle recognition based on Faster R-CNN. The fundamental convolution layer of the Faster R-CNN is composed of two convolution layers built using Mobile Net technologies. The soft-NMS method, which replaces NMS after the region proposal network in the original Faster R-CNN, provides greater accuracy and prevents duplicate proposals. An ROI pooling layer that takes account of contextual information is then used to modify the proposals to the desired size without losing any of the contextual information.[15] To categorize proposals and change the bounding box of observed vehicles, the Faster R-CNN framework's final stage uses a depth-wise separable convolution structure from the Mobile Net architecture.

An improved SSD model-based front vehicle detection approach for smart automobiles is presented by Cao et al. [16]. One of the most common deep learning-based object recognition frameworks nowadays is the Single Shot multi-box Detector (SSD). This study begins with a brief introduction to the SSD network [17] concept, followed by an examination and explanation of its flaws and deficiencies in vehicle recognition.

Gao et al. [18] suggested En-RetinaNet, a bottom-up upgraded RetinaNet model for superior vehicle recognition performance. Before the region proposal network, the EnRetinaNet contains an upgraded Feature Pyramid Network (FPN) and a bottom-top fusion. A bottom layer is added to the feature pyramid network in the modified feature pyramid network to utilize more local features

3. METHODOLOGY

Figure 2 depicts the proposed system's overall architecture, which comprises the following stages: a) For Data Collection, we used IVS-1 and IVS-2 datasets b) Fusion of IR-VI images using Auto-encoder network c) vehicle detection using Yolov3 network d) Feature extraction where ORB algorithm is used e) vehicle classification for classifying those vehicles based on image instances.

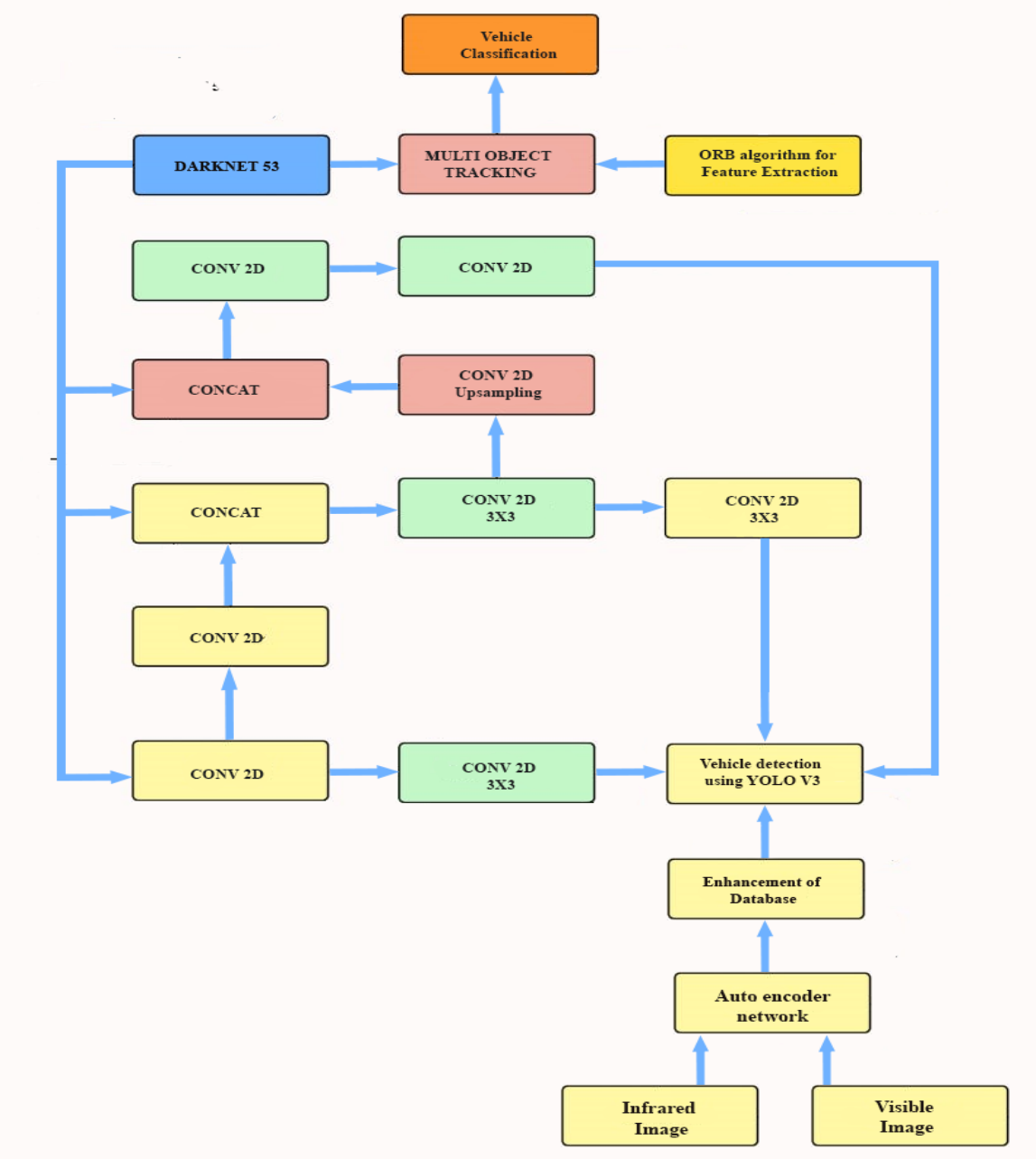

Figure 2. The overall architecture of the proposed fusion, detection and classification

3.1 Data Collection

When applying machine learning approaches to address object detection difficulties, datasets are critical. We gather a large number of open-source datasets from the web, correct certain incorrect bounding boxes, and convert them to XML format. In addition to the open-source datasets (called IVS-2 dataset), we also establish our datasets, acquired by the Papago P1W Carmax, a 120°from-view automobile event recorder. The datasets utilised to train the proposed model are listed in Table 1. IVS-1 is a dataset that represents the depression angle perspective. In comparison to the IVS-1 dataset, the IVS-2 dataset has the polar opposite view angle (dashcam view). The IVS-1 dataset is primarily used to fine-tune the model because it is for the intended applications. To improve vehicle type discrimination, we additionally employ other datasets [19,20].

Table 1: Dataset of vehicles in the proposed system

Dataset Name	Number of images
IVS-1 (depression angle view)	316734
IVS-2 (dashcam view)	599278

The first step is detecting the target vehicles with maximum accuracy, and then we classify them into different types, including large-sized cars, trucks, cars, motorcycles, and bicycles, in order to establish the new dataset. Table 2 shows the total number of images available: 90,920 trucks, 604,153 sedan/SUV images, 54,050 bus images, 26,568 van images, 138,525 scooter images, and 1,801 bike images. During the construction of the dataset, the balance of these desired elements must be maintained. This is intrinsically tied to the quality of object identification and categorization. The number of motorcycles in the sample should be raised as much as possible, according to Table 2.

Table 2. Target objects categories

Car type	IVS-1 dataset	IVS-2 dataset	Total
Truck	10263	80095	90920
Trailer-head	562	80095	90920
Sedan/SUV	220094	384059	604153
Bus	29641	24409	54050
Van	6055	20513	26568
Scooter	48321	90204	138525
Bike	1801	0	1801

3.2 Fusion of IR and VI images

By utilizing the differences in radiation, infrared images allow us to differentiate targets from their backgrounds, regardless of whether night/day conditions. Visible images, on the other hand, can convey textural details with great spatial resolution and definition in a way that mimics the human visual system. In order to combine the advantages of infrared and visible images, it makes sense to fuse the two types of images. Infrared images provide information about thermal radiation while visible images provide information about texture.[21]. Because of deep learning's recent rise in popularity, deep learning algorithms have been employed to address the fusion of IR and VI images. We propose an auto encoder network to fuse infrared & visible images.

Convolutional Neural Network (CNN) [13]is a popular deep learning model that seeks to develop an effective feature representation method for image at several abstraction levels. CNN model can successfully represent the spatial and temporal relationships in an image through the use of appropriate filters. Unlike other applications, feature collection or activity level measurement is the major concern while selecting the CNNs for image fusion problems. Maxpooling is useful for reducing the amount of data to decrease the computational complexity. Researchers in multifocus image fusion are focusing on the deep focus features but maxpooling results in information loss. To overcome this problem, the layers are reformulated to learn from residual connections as well as the layer inputs. Experimental results with residual network used for learning reduces problems with overfitting and other training error. Deep residual nets are very easy to optimize, and they can quickly benefit with increasing depth, providing results that are far better than earlier networks.

Here, we adopt a two-branch scheme in which the top branch uses few residual connections in inner layers with a filter size of 3x3 for collecting the deep features. The bottom branch also collects some fine features and the filter size selected for the feature extraction is 5x5.16 filters are selected in every CNN layer to extract the relevant features. Output generated by each layer is given to the successive layer to collect deep features for focus map generation. Residual networks reduce the chances of overfitting of the model and helps in the successful convergence of the training algorithm. Feature extraction process is depicted in Figure 3. The proposed auto encoder network is trained to optimize the feature collection procedure. Features collected by encoder is combined to generate 80 feature maps for fusion. Feature maps are given to the decoder network to reconstruct image from features. Decoder network consist of 5 CNN layers with selected kernel of 3x3. An L2 norm fusion strategy is adopted select the deep features for the fusion. Infrared and vehicle images are given to the trained model and the feature maps are combined to form the fused feature map. Fused feature map is given to the reconstruction network to generate fused image.

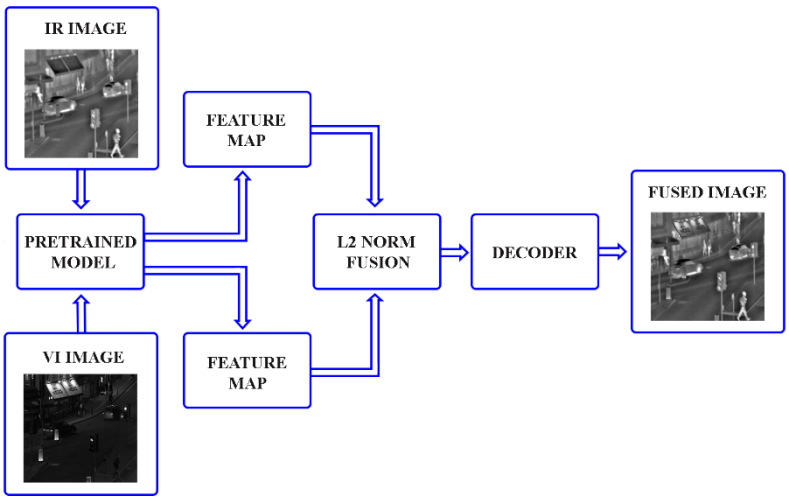

Figure 3.Proposed fusion Procedure

Figure 4 shows the deep autoencoder network architecture used for fusing infrared and visible images and Figure 5 shows some fused outputs generated by the fusion model.

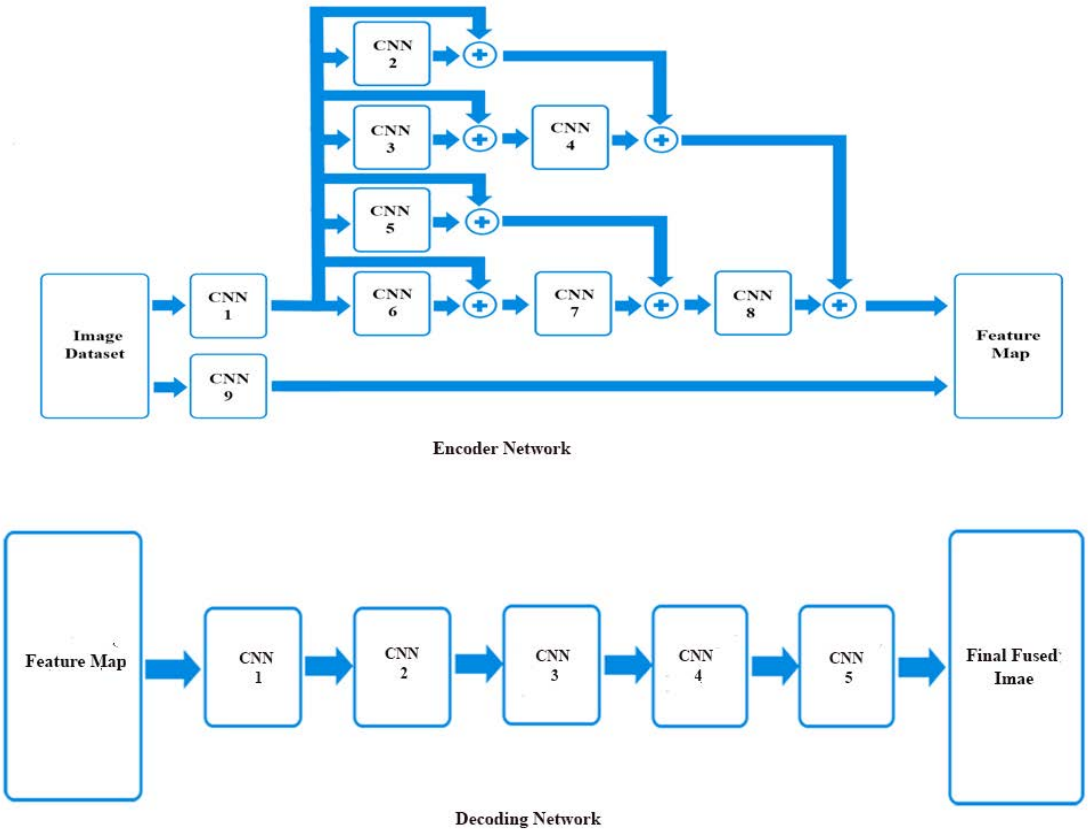

Figure 4. Auto – encoder Architechture

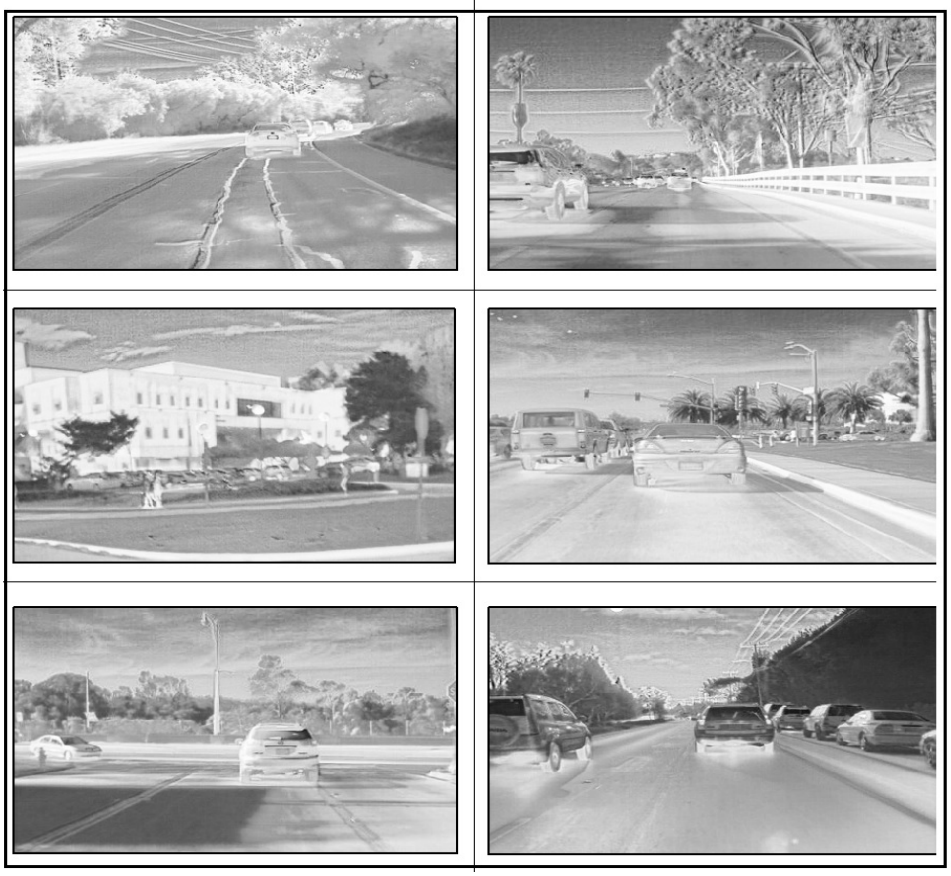

Figure 5. Image instances of combining visible and IR images using fusion

3.3Vehicle Detection: Yolov3

A unified object detection paradigm called YOLO (You Only Look Once) was suggested [22]. Detection is viewed as a regression problem in YOLO. It's a pre-trained model that doesn't require any training data. Weights make up the system, and boxes are used to detect objects. The model regresses the image to tensor, revealing the digit of every object's position as well as the object's class score. The photographs that are entered do not have to pass through the YOLO network again. As a result, picture processing in this model is faster. Yolo's object detection accuracy is more than 50 times higher than other object detection models. As a result, YOLO has evolved into one of the most powerful real-time object recognition technologies in the market. In real-time, the classic YOLO model can process up to 45 frames per second, whereas Fast YOLO models can process up to 155 frames per second. The miniature variation is the network's basic version. This approach may be used to generalize real pictures extremely successfully. [23,24].

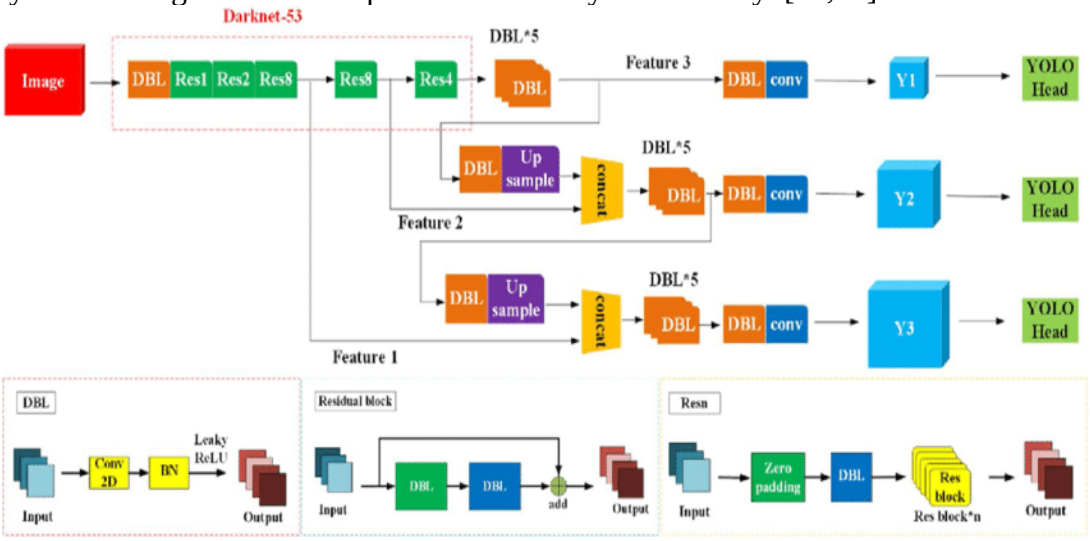

Figure 6. Yolo V3 architecture of the proposed approach

Object detection may be done in two ways. The initial step is to use object photographs to build our own Deep Learning model. Features are the most important input for the machine learning model. The model learns and generates weights for the item based on its characteristics. This technique, however, has several disadvantages. When we study the object of a car, for example, we may divide it into different categories based on its shape. In rare circumstances, a truck may appear in the video as a vehicle. All components of the model, such as size, dimension, and form, should be trained. This necessitates a vast amount of data. As a result, our system will be used for training. If the GPU on the machine isn't powerful enough, we will be unable to train our model at all, or it will take an inordinate amount of time to do so. As a result, we utilize a YOLO model (Figure 6), which is characterized by the convolution neural network idea. YOLO differs from previous CNN models in that it includes a moving or floating window. This signifies that a left-to-right moving window is formed in YOLO. If a necessary object arrives on the screen while you're moving, YOLO will highlight it. It will attempt to identify and recognise the item using the weights already contained in the model. In YOLO, there is a distinct weight for each object. YOLO comes in a variety of forms. Some models may have 150 items, while others may only have 80. The number of items can be limited to whatever is necessary, or all of the objects in the model can be used. A model with 80 weights was employed in this study. We must load the weights every time the code runs. We limit the weights for the first 10 since we don't want our model to recognize all 80 weights or items. Object detection is handled in this way in YOLO.

3.4 Vehicle Tracking

Although TensorFlow is being used to detect the object, it does not track it. To track the object, bounding boxes are given to each object on the screen or in the video. The Kernel Dimension of the Weights is delivered by TensorFlow. Four coordinates are extracted from the eight coordinates of the kernel dimension. We double the width and height of the kernel dimension coordinates to adjust the dimension of the item that has been recognized.

The characteristics of the identified automobiles were extracted using the ORB algorithm, and satisfactory results were achieved. The ORB approach exceeds the competition in terms of processing efficiency and matching expenses. The SIFT and SURF image description methods can be replaced with this strategy. Using the features from the Accelerated Segment Test, the ORB approach discovers feature points before using the Harris operator to identify corners (FAST). Once the feature points have been gathered, the descriptor is computed using the BRIEF approach. The coordinates are calculated by utilizing the centroid of the point region as the coordinate system's x-axis and the feature point as the circle's center. As a result, if we rotate the image, the coordinate system will rotate as well, ensuring that the feature point descriptor rotates appropriately. When the picture angle is changed, a consistent point can also be proposed. The XOR technique is used to match the feature points after acquiring the binary feature point description. This enhances the efficiency of matching [25,26]

Figure 7. Features extracted by ORB algorithm for vehicle detection

To extract the object features from the object detection box, the ORB algorithm was used. The extraction of the object features does not depend on the entire road surface, thereby reducing computation time dramatically. Because the vehicle object moves very little in a continuous frame of video, the ORB feature obtained in the object box is used to generate the prediction box of the object in the next frame in object tracking. If the prediction box and detection box in the following frame fulfil the center point's least distance criteria, the same item successfully matches between the two frames (Figure 7). T is the pixel difference between two video frames with the highest variation in the observed center point of the vehicle object box. In the next two frames, the vehicle's positional displacement is less than T. As a result, when the vehicle object box's centre point crosses T twice in a row, the automobiles in the two frames don't match, causing the data association to fail. The size of the vehicle object box is related to the T threshold value while analyzing scale change during vehicle movement. Different criteria apply to the vehicle object boxes. This standard might support vehicle mobility as well as a variety of video input sizes. The height of the vehicle object box is given by Eq.1, and T is determined from it.

$$T = \text{box height} / 0.25 \quad (1)$$

3.5 Vehicle Classification

To assess vehicle categorization abilities, a visual classifier based on the YOLO approach is utilized. The YOLO technique is used to categorize each vehicle into one of six groups using the visual classifier in Figure 6. During the training phase, when a vehicle from one of the six classes is spotted, all bounding boxes are collected, classes are manually labelled, and the labelled data is put into the YOLO model for classification.

The YOLOv3 model architecture is employed in this work, as shown in Figure 6.The Darknet-53 network receives images at a resolution of 416 x 416 pixels. Darknet-53 [27,28] is the name given to a feature extraction network with 53 convolutional layers. Alternating convolution kernels are used in Darknet-53, with a batch normalizing layer applied after each convolution layer for normalization. The feature map's size is reduced by removing the pooling layer and increasing the convolution kernel step size. The YOLOv3 model extracts feature using ResNet before building three features with sizes of 13x13x1024, 26x26x512, and 52x52x256 pixels using the feature pyramid top-down and lateral connections. The final output depth is (5 + class) x3, showing that four basic parameters, as well as the credibility of a box across three regression bounding boxes and the likelihood of each class is included within the bounding box, are all predicted. Each class is scored using the sigmoid function from YOLOv3. The item is classified as belonging to a specific category when the class score exceeds a certain threshold, and each object can have many class identities that do not conflict with one another.

4.Performance Evaluation

The proposed system (Yolo-ORB) is developed using software specifications like Pytorch as the programming language which is an open-source library inside python for deep learning purposes. The GTX 1050 Ti 4GB Graphics (Core i7-8750H 8th Gen/8GB RAM/1TB SSHD + 128GB SSD) and Windows 10 OS are used to create this model. The proposed model is evaluated using measures like accuracy, sensitivity, specificity, recall, precision, F1-score, detection rate, TPR, FPR, IoU, mAP, computation time and memory utilization over models like RCNN [7], VGG16[29], Alexnet[30], Googlenet[31], CNN, Faster-RCNN[14], Yolov3-tiny[32]. For fusion related comparison with respect to the proposed system (Siamese-CNN) with other models like GAN-based, CNN-based, NSCT, The Correlation Coefficient (CC)[33] is the most accurate. PCNN outperforms metrics such as the Correlation Coefficient (CC), Peak Signal-to-Noise Ratio (PSNR) [34], and Structural Similarity Index Measure (SSIM) [35].

Table 3. Overall analysis under accuracy, sensitivity, specificity, precision and recall

Models	Accuracy	Sensitivity	Specificity	Precision	Recall
RCNN	88	89	91	88	75
VGG16	84	88	92	79	71
Alexnet	81	89	94	82	76
Googlenet	86	90	94	85	78
CNN	91	94	96	92	80
Faster-RCNN	93	95	97	94	81
Yolov3-tiny	95	97	97	95	80
Yolov3-ORB(Ours)	97	98	97	96	82

Table 3 shows the overall analysis of several models using metrics such as accuracy, sensitivity, and specificity. The proposed method outperforms the other models with 0.97 accuracy, 0.98 sensitivity and 0.97 specificity. Other models also improve their performance too but yolo-tiny and the proposed system (yolov3-ORB) have a more

complex structure and the analytical pattern brings more effectiveness overall. Our method also shows good performance with 0.95 precision and 0.82 recall over other models.

Table 4 depicts the overall analysis of various models under measures like detection rate, TPR and FPR. The proposed method outperforms the other methods with 0.84 F1-score,0.94 detection rate, 0.91 TPR and 0.9 FPR over other models.

.

Table 4. Overall analysis under F1-score, Detection rate, TPR and FPR

Models	F1-score	Detection rate	TPR	FPR
RCNN	78	85	81	19
VGG16	70	86	82	18
Alexnet	74	82	78	22
Googlenet	76	88	85	15
CNN	78	91	88	12
Faster-RCNN	81	92	89	11
Yolov3-tiny	83	93	90	10
Yolov3-ORB(Ours)	84	94	91	9

Table 5 gives the analysis all the models considered against iOU and mAP values. Our method also gave good values for 0.5 iOU and 0.98 mAP over other models**.**

Table 5. Overall analysis under IoU and mAP

Models	IoU	mAP
RCNN	0.5	0.81
VGG16	0.5	0.73
Alexnet	0.5	0.85
Googlenet	0.5	0.89
CNN	0.5	0.92
Faster-RCNN	0.5	0.93
Yolov3-tiny	0.5	0.95
Yolov3-ORB(Ours)	0.5	0.98

Table 6. The overall analysis of fusion models under CC, PSNR and SSIM

Models	CC	PSNR	SSIM
GAN-based	0.71±24.2	0.73±30	0.68±41
CNN-based	0.81±14	0.83±20	0.75±10.8
NSCT	0.79±33.1	0.80±27	0.71±12
PCNN	0.75±22	0.84±17	0.77±21
Auto encoder model	0.84±4.1	0.88±20	0.80±9.2

Table 6 depicts the overall analysis of fusion methods with our autoencoder model over measures CC, PSNR and SSIM. The proposed method outperforms the other models with 0.84±4.1 CC, 0.88±10.2 PSNR, and 0.80±9.2 SSIM. values. Figure 8 shows some results of the vehicle detected using proposed model. More vehicles are detected due to the presence complementary information from VI and IR images.

Figure 8. Vehicle Detection with bounding box using Yolov3 for certain image instances

5. CONCLUSION

In this paper, the importance of deep learning and as well as the class imbalance of dataset in evaluating a model to give effective results over vehicle detection is stated. An effective model is been proposed by integrating a deep learning model Yolov3, an advanced and complex network for detecting and also classifying the vehicles along with the extraction algorithm ORB. We also inserted a fusion algorithm using autoencoder network which also brings effective results in casting the VI-IR images. Proposed system has better accuracy over small objects, a fast detector at high speed, considerable time for training, additional algorithms that compensates detection rate and so it is similar to latest version of Yolo. Evaluation of the proposed model under various measures and also evaluating fusion model with their respective indexes brings a clear idea of the accurate model and also this paper will help other researcher specialists to dig deep for analyzing and bring even more advanced integrated concepts.

Acknowledgments

H. Shihabudeen would like to thank the College of Engineering Thalassery, College of Engineering Kidangoor, and Centre for Engineering Research and Development (CERD) APJ Abdul Kalam Technological University, Kerala, for giving support for carrying out the research work. He also thanks his supervisor, who has provided many directions in conducting this research.

References

[1] L. Qi, "Research on Intelligent Transportation System Technologies and Applications, " in 2008 Workshop on Power Electronics and Intelligent Transportation System, Aug. 2008, pp. 529-531.

[2] S. Huang, Y. He, and X. Chen, "M-YOLO: A Night time Vehicle Detection Method Combining Mobilenet v2 and YOLO v3," vol. 1883, p. 012094, Apr. 2021,

[3] G. Xiong, X. Shang, X. Liu, and J. Cao, "Chapter 8 - Novel ITS based on space-air-ground collected Big Data," in Big Data and Smart Service SystemsEds. Academic Press,2017, pp. 115-137.

[4] "TensorFlow Lite - Real-Time Computer Vision on Edge Devices (2022),"viso.ai, Sep. 21, 2021. https://viso.ai/edge-ai/tensorow-lite/ (accessed Apr. 01, 2022).

[5] J. Azimjonov and A. Ozmen, "A real-time vehicle detection and a novelvehicle tracking systems for estimating and monitoring traffic flow on highways," Adv. Eng. Inform., vol. 50, p. 101393, Oct. 2021,

[6] S. Javadi, M. Dahl, and M. I. Pettersson, "Vehicle Detection in Aerial Images Based on 3D Depth Maps and Deep Neural Networks," IEEE Access, vol. 9, pp. 8381-8391, 2021,

[7] A. A. Yilmaz, M. S. Guzel, I. Askerbeyli, and E. Bostanci, "A Vehicle Detection Approach using Deep Learning Methodologies,"ArXiv180400429 Cs, Apr. 2018, Accessed: Apr. 04, 2022.

[8] J.-G. Wang et al., "Appearance-based Brake-Lights recognition using deep learning and vehicle detection," in 2016 IEEE Intelligent Vehicles Symposium (IV), Jun. 2016, pp. 815-820.

[9] B. Major et al., "Vehicle Detection With Automotive Radar Using Deep Learning on Range-Azimuth-Doppler Tensors," in 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW), Oct. 2019, pp. 924-932.

[10] J.-S. Shin, U.-T. Kim, D.-K. Lee, S.-J. Park, S.-J. Oh, and T.-J. Yun, "Real-time vehicle detection using deep learning scheme on embedded system," in 2017 Ninth International Conference on Ubiquitous and Future Networks (ICUFN), Jul. 2017, pp. 272-274.

[11] J. E. Espinosa, S. A. Velastin, and J. W. Branch, "Vehicle Detection Using Alex Net and Faster R-CNN Deep Learning Models: A Comparative Study," in Advances in Visual Informatics, Cham, 2017, pp. 3-15.

[12] S.-C. Hsu, C.-L. Huang, and C.-H. Chuang, "Vehicle detection using simplified fast R-CNN," in 2018 International Workshop on Advanced Image Technology (IWAIT), Jan. 2018, pp. 1-3.

[13] K. Shi, H. Bao, and N. Ma, "Forward Vehicle Detection Based on Incremental Learning and Fast R-CNN," in 2017 13th International Conference on Computational Intelligence and Security (CIS), Dec. 2017, pp. 73-76.

[14] H. Nguyen, "Improving Faster R-CNN Framework for Fast Vehicle Detection," Math. Probl. Eng., vol. 2019, p. e3808064, Nov. 2019,

[15] L. Sommer, A. Schumann, T. Schuchert, and J. Beyerer, "Multi Feature Deconvolutional Faster R-CNN for Precise Vehicle Detection in Aerial Imagery,"

in 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), Mar. 2018, pp. 635-642.

[16] J. Cao et al., "Front Vehicle Detection Algorithm for Smart Car Based on Improved SSD Model," Sensors, vol. 20, no. 16, Art. no. 16, Jan. 2020,

[17] S. N. Ferdous, M. Mostofa, and N. M. Nasrabadi, "Super resolution assisted deep aerial vehicle detection," in Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications, May 2019, vol. 11006, pp. 432-443.

[18] P. Gao, J. Tian, Y. Tai, T. Zhao, and Q. Gao, "Vehicle Detection with Bottom Enhanced RetinaNet in Aerial Images," in IGARSS 2020 – 2020 IEEE International Geoscience and Remote Sensing Symposium, Sep. 2020, pp. 1173-1176.

[19] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation,"ArXiv13112524 Cs, Oct. 2014,

[20] L. Yang, P. Luo, C. C. Loy, and X. Tang, "A large-scale car dataset for fine-grained categorization and verification," in 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Jun. 2015, pp. 3973-3981.

[21] H. Shihabudeen and J. Rajeesh, "Deep Learning L2 Norm Fusion for Infrared & Visible Images," in IEEE Access, vol. 10, pp. 36884-36894, 2022. doi: 10.1109/ACCESS.2022.3164426.

[22] B. Benjdira, T. Khursheed, A. Koubaa, A. Ammar, and K. Ouni, "Car Detection using Unmanned Aerial Vehicles: Comparison between Faster R-CNN and YOLOv3,"in 1st International Conference on Unmanned Vehicle Systems-Oman (UVS), Feb. 2019, pp. 1-6.

[23] X. X. Zhang and X. Zhu, "Moving vehicle detection in aerial infrared image sequences via fast image registration and improved YOLOv3network," Int. J. Remote Sens., vol. 41, no. 11, pp. 4312-4335, Jun. 2020,

[24] L. Zhou, J. Liu, and L. Chen, "Vehicle detection based on remote sensing image of Yolov3," in 2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), Jun. 2020, vol.1, pp. 468-472.

[25] Yang, S. Zhang, Y. Tian, and B. Li, "Front-Vehicle Detection in Video Images Based on Temporal and Spatial Characteristics," Sensors, vol. 19,no. 7, Art. no. 7, Jan. 2019,

[26] M. Kumar, S. Ray, D. K. Yadav, and R. Tanwar, "A Study of Moving Vehicle Detection and Tracking Through Smart Surveillance System,"in Proceedings of International Conference on Intelligent Cyber-Physical Systems, Singapore, 2022, pp. 303-315.

[27] C. R. Rashmi and C. P. Shantala, "Vehicle Density Analysis and Classification using YOLOv3 for Smart Cities," in 2020 4th International Conference on Electronics, Communication and Aerospace Technology(ICECA), Nov. 2020, pp. 980-986.

[28] P.-Y. Sun, W.-Y. Sun, Y. Jin, and R. O. Sinnott, "Heavy Vehicle Classification Through Deep Learning," in Big Data - BigData 2020, Cham, 2020, pp. 220-236.

[29] K. Simonyan and A. Zisserman, "Very Deep Convolution Networks or Large-Scale Image classification", Computer Science, 2014.

[30] A. Krizhevsky, I. Sutskever and G. Hinton, "Imagenet classification with deep convolutional neural networks", Advances in Neural Information Processing Systems 25, pp. 1106-1114,

[31] C. Szegedy et al., "Going deeper with convolutions," 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp.1-9.

[32] P. Adarsh, P. Rathi and M. Kumar, "YOLO v3-Tiny: Object Detection and Recognition using one stage improved model," 2020 6th International Conference on Advanced Computing and Communication Systems(ICACCS), 2020, pp. 687-694

[33] Bai X. Morphological center operator based infrared and visible imagefusion through correlation coe_cient. Infrared Phys Technol. 2016;76:546-55

[34] F.E. Ali, I.M. El-Dokany, A.A. Saad & F.E. Abd El-Samie (2010) A curvelet transform approach for the fusion of MR and CT images, Journal of Modern Optics, 57:4, 273-286

[35] H. Shihabudeen and J. Rajeesh, "Euclidian Norm Based Fusion Strategy for Multi Focus Images," 2021 2nd International Conference on Advances in Computing, Communication, doi: 10.1109/ACCESS51619.2021.9563338.