I.J. Image, Graphics and Signal Processing, **2023, 5, 17-28**
Published Online on October 8, 2023 by MECS Press (http://www.mecs-press.org/)
DOI: 10.5815/ijigsp.2023.05.02

Real-Time Vehicle Detection for Surveillance of River Dredging Areas Using Convolutional Neural Networks

Mohammed Abduljabbar Zaid Al Bayati
Karabuk University / Computer Engineering, Karabuk, Turkey
Email: mohameed.nkb.1991@gmail.com
ORCID iD: https://orcid.org/0000-0002-2535-1742

Muhammet Çakmak*
Karabuk University / Electrical and Electronics Engineering, Karabuk, Turkey
Email: muhammetcakmak@karabuk.edu.tr
ORCID iD: https://orcid.org/0000-0002-3752-6642
*Corresponding Author

Received: 21 April 2023; Revised: 27 May 2023; Accepted: 06 July 2023; Published: 08 October 2023

Abstract: The presence of illegal activities such as illegitimate mining and sand theft in river dredging areas leads to economic losses. However, manual monitoring is expensive and time-consuming. Therefore, automated surveillance systems are preferred to mitigate such activities, as they are accurate and available at all times. In order to monitor river dredging areas, two essential steps for surveillance are vehicle detection and license plate recognition. Most current frameworks for vehicle detection employ plain feed-forward Convolutional Neural Networks (CNNs) as backbone architectures. However, these are scale-sensitive and cannot handle variations in vehicles' scales in consecutive video frames. To address these issues, Scale Invariant Hybrid Convolutional Neural Network (SIH-CNN) architecture is proposed for real-time vehicle detection in this study. The publicly available benchmark UA-DETRAC is used to validate the performance of the proposed architecture. Results show that the proposed SIH-CNN model achieved a mean average precision (mAP) of 77.76% on the UA-DETRAC benchmark, which is 3.94% higher than the baseline detector with real-time performance of 48.4 frames per seconds.

Index Terms: River dredging, Automated surveillance, Vehicle detection, CNN, Scale invariant

1. Introduction

Vehicle detection mechanism playing a pivotal role in vision-based surveillance systems, such as vehicle theft detection[1] , speed enforcement [2], automatic non-stop tolling [3], intelligent traffic observation [4], and illegal activity recognition [5], among others. In the realm of water conservation projects, river dredging assumes a significant role. It plays a crucial part in maintaining water resources, preventing floods, and enhancing water flow [6]. River dredging is a major part of water conservation projects [7]. The extraction and sale of gravel and sand during river dredging contribute to considerable economic [8]. However, this essential process is not without its challenges, as illegal mining and sand theft [9] pose serious security threats to the hydraulic engineering department. Consequently, continuous surveillance of dredging areas has become a critical responsibility for relevant government agencies.

Every day, at river dredging construction sites where hundreds of vehicles enter and hundreds of vehicles traverse these dredging sites, manual inspection by patrol officers or security guards are time-consuming, costly, and laborious for humans. The introduction of an automated vision-based monitoring system enables the accurate detection and recognition of vehicles. This innovation significantly reduces the risk of compromised inspections resulting from bribes or threats posed by unauthorized entities. Furthermore, the autonomous system not only heightens the security and safety of the construction area but also optimizes the inspection process, thereby avoiding potential disruptions to the construction timeline [5]. Consequently, the real-time vehicle detection provides a robust foundation for improved management at control checkpoints and ensures more efficient surveillance. According to literature analysis, vehicle detection is grouped into two categories: (1) traditional Machine Learning (ML)-based and (2) deep learning-based [10]. In ML-based models, initially a feature extraction mechanism is applied to extract features, and then a classifier is

utilized to categorize these extracted features [11,12]. In the case of DL, especially in Convolutional Neural Network (CNN)-based model, there is no need to extract hand-crafted features [13,14].

The CNN-based model automatically extracts low-level features and trains the model using the most optimal features [11,15]. CNN-based architectures have played a vital role in vision-based object recognition [16-21]. However, there are some issues with the identification of vehicles. Firstly, vehicles are challenging to detect because of their diversified shapes, hues, and sizes. Secondly, the geometry of vehicles in successive video frames changes based on their position and orientation. Thirdly, environmental variables may have an impact on the outcomes of vehicle detection. Lastly, vehicle recognition applications demand real-time performance and real-time detection system is essential for vehicle recognition applications.

Many vision-based detection techniques have already been proposed and discussed in the literature. The object detectors described in [16-21] are used in many vehicle surveillance models. But these detectors are designed for object detection from a single image. In River Dredging Areas (RDA), surveillance cameras capture vehicle at different scales. If we directly apply these detectors to RDA applications, they neglect multiscale features that are valuable to detect small sized vehicles.

In the context of river dredging area vehicle detection, vehicles constantly change in scale as they move through the surveillance area. This variability in vehicle size becomes particularly evident in successive video frames captured by road supervision cameras. When vehicles are farther from the camera, they appear smaller in the image, and as they get closer, they occupy a larger area as shown in Fig.1. Consequently, even if the vehicle type remains same, its scale may differ significantly in consecutive frames. This variability complicates the accurate and reliable detection of vehicles. Detectors proposed in [19,21] utilized grid cells for detection, the grid size played a vital role in vehicle detection accuracy and time complexity. These schemes divides the image into (7×7) grid cells, these larger grid size are computationally efficient but fail at small scale vehicle detection [22].

Fig. 1. Scale change in consective video frames

For vehicle detection in RDA, due to the large sized grid cells, detectors [19,21] will failed to detect when vehicles far away from cameras and appear smaller. Moreover, the detector presented in [21] employed DarkNet-19 as architecture. In the quest for cutting-edge vehicle detection, DarkNet-19, with its simple feed-forward CNN architecture, falls short of the mark. The absence of multiscale and multilevel descriptors proves detrimental, causing gradient vanishing/exploding and undermining the network's ability to handle class variation effectively [23]. With the automotive landscape constantly evolving, relying on DarkNet-19 alone would be akin to navigating a winding road blindfolded [22]. To overcome these challenges and pave the way for more accurate and reliable vehicle detection, our research advocates for the integration of a more sophisticated neural network one that harnesses the power of multi-level features, ensuring smooth and seamless identification of vehicles, regardless of their shape, size.

In this study, a vehicle detection system is designed for surveillance applications in river dredging areas while taking into account the constraints of existing frameworks. The proposed system, as shown in Fig. 2, takes video frames of vehicles at the entrance point. After pre-processing, the proposed vehicle detector is applied to detect vehicles for vehicle recognition mechanisms that facilitate the government’s agencies in identifying illegal activities.

The contributions of this research paper summarized as follows:

- In order to address the challenge of scale variation in moving vehicles, we have proposed a Scale Invariant Hybrid Convolutional Neural Network (SIH-CNN) architecture. This approach allows for improved detection accuracy by effectively handling the varying sizes of vehicles in consecutive frames. The SIH-CNN architecture is designed to be scale-invariant, enabling it to adapt to the changing scales of vehicles, resulting in more reliable and accurate vehicle detection.
- To mitigate the issue of gradient vanishing and enhance the model's ability to handle class variation, we have introduced a multi-level feature extraction block. This component plays a crucial role in improving the overall

robustness of the detection system. By effectively extracting hierarchical features from the input data, the multi-level feature extraction block helps capture essential information at different abstraction levels, leading to better detection performance.

- In order to enhance the detection of tiny vehicles, we have incorporated a multi-scale feature extraction block into our proposed scheme. Small-sized vehicles often pose a challenge for traditional detection methods [22]. By extracting features at different scales, this block allows the model to detect and recognize small vehicles more accurately, contributing to an overall improvement in the performance of the vehicle detection system.

In conclusion, our research paper presents a comprehensive approach to address the scale change of travelling vehicles in consecutive video frames. By leveraging the Scale Invariant Hybrid Convolutional Neural Network (SIH-CNN) architecture, along with the multiscale and multilevel descriptors extraction blocks, we have achieved significant advancements in detection accuracy, robustness, and the ability to detect small-sized vehicles. Moreover, High frame rates, approximately 30 FPS, are essential for accurately capturing and recognizing moving vehicles [24]. Our proposed SIH-CNN operates at an impressive 48.4 frames per second, thereby making it highly efficient and well-suited for real-time surveillance applications. These contributions pave the way for more effective and reliable vehicle detection systems in real-world applications, such as river dredging area monitoring, where accurate vehicle detection is of utmost importance.

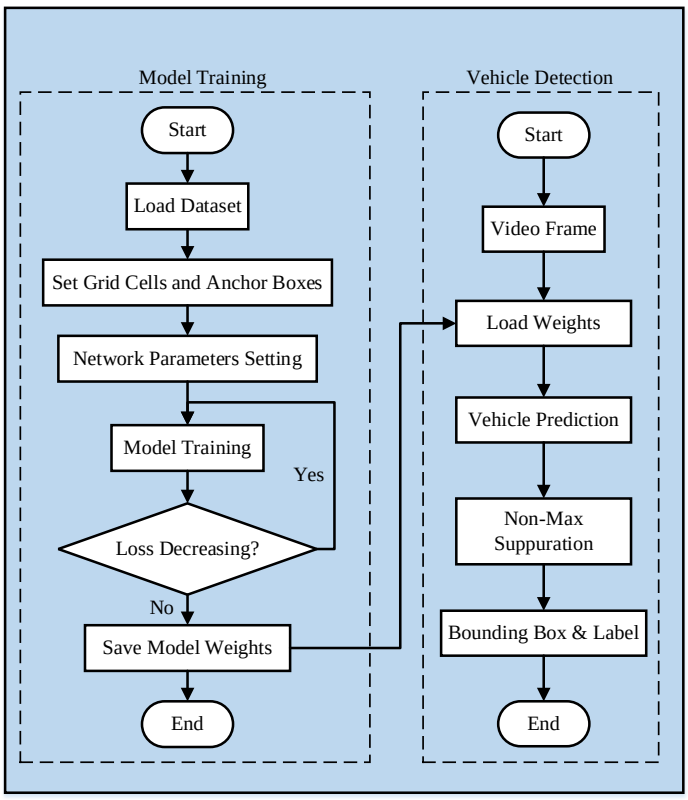


Fig. 2. Vehicle Detection Process

The rest of the paper is categorized into different sections: Section 2 discusses the current state-of-the-art vehicle detection frameworks; Section 3 describes the proposed methodology, where we discuss the detection mechanism, proposed backbone architecture, and proposed loss function. Section 4 discusses the experiments, which cover the experimental environment, benchmarks, performance evaluation metrics, and quantitative and qualitative results to validate the system's performance. Finally, Section 5 described the conclusion and future work of this investigation.

2. Literature Review

Vehicle detection is considered as an elementary step of traffic survellince applications. And deep convolutional neural networks (DCNN) have achieved tremendous success in vision-based detection with the evolution of domain-specific architectures (DSA). Most of the vehicle detection frameworks are based on these detection models [16-21,25].

In [26], the authors proposed a vehicle detection framework based on Faster-RCNN. Their proposed framework is specially designed for dense background conditions. In a single framework, they used two networks (the fine-tuning

network and the proposal network). They utilized VGG-16 for features extraction and training. The VGG-16 network ignored the multilevel descriptors that are suitable to handle vehicle class variation.

In [27], Hu et al. proposed a vision-based vehicle detection framework. Their proposed framework utilised context-aware Region-of-Interest (RoI) pooling to produce accurate feature maps. They utilised a multi-branch decision network for vehicle classification. The VGG and PVANET CNN backbones are used to train the proposed framework. Their backbone architectures are simple feed-forward neural networks that ignore the vast majority of valuable semantic information, and suffer from detection errors or misses.

In [28], authors proposed a fast vehicle detection framework for traffic surveillance. They introduced "connect and merge residual networks" to enhance classification accuracy. They also designed a multi-scale prediction network to precisely predict the shape of vehicles. They utilised the Darknet-19 backbone architecture that is used in YOLO V2 [21]. The Darknet-19 ignores the multilevel descriptors, and the pooling in descending layers of the Darknet-19 that miss detecting small-scale vehicles [22].

In [29] authors proposed a CNN based vehicle detection method to handle occlusion of vehicles. The proposed scheme used K-means to cluster the aspect ratio and vehicle scale in the dataset. Low- and high-level features are concatenated using feature fusion techniques, and these features are used to detect vehicles. But their proposed model ignored the small-scale vehicle with a height of less than twenty pixels.

Table 1. Literature analysis

Detection Algorithms		Contribution	Limitations	Accuracy	Sensitivity	Complexity
Two-Stage Detectors	R-CNN [16]	Bypass the issue of selecting large number of regions and utilize Selective Search Algorithm (SSA) for region proposal then apply CNN based classifier to classify these regions.	Selective Search results in bottleneck. The detection speed is slow 47 seconds per image. Not suitable for real time applications.	Low	Low	Medium
	Fast-RCNN [17]	Introduce the convolutional feature map instead of SSA (i.e., used in R-CNN).	Not appropriate for real time applications. The detection speed is time-consuming only 5-frames per seconds.	Low	Low	Medium
	Faster-RCNN [18]	Introduce Region-Proposal-Network (RPN) that replace SSA for region proposal.	Proposed detector is position sensitive and translation invariant. And detection speed not good as single stage detectors.	Medium	Medium	Medium
	R-FCN [34]	Introduce the position-sensitive score maps to overcome the problems of Faster -RCNN.	Proposed detector is not suitable for real time detection problem.	High	Low	High
Single-Stage Detectors	YOLO [19]	YOLO is first single stage detector, that introduced regression-based detection []. Proposed detectors classify and localize objects simultaneously.	The detection accuracy is very low. Large grid cells result in miss detection of small-scale vehicles. Poor localization.	Low	Medium	Low
	SSD [20]	Introduced more anchor points to precise localization. Utilize multi scale features to enhance detection accuracy for small scale vehicles.	Poor detection accuracy specially for small scales vehicles when vehicles are away from the cameras.	Low	Medium	Low
	YOLO V2 [21]	Introduced multi-scale training, adoptive anchor boxes and proposed Darknet19 a CNN based backbone architecture.	Proposed model cannot handle scale variation of vehicles in consecutive video frames [22].	Low	High	Low
	HVD-Net [[22]]	Introduce multilevel and multiscale features to handle gradient vanashing and class variation problems.	DSPP becomes ineffective if the window size increases to a large number [33]. Moreover DSPP differnt size windows increase model complexity and impede end-to-end training.	High	Medium	High
	Retina-Net [25]	Proposed a single stage detector with focal loss, to resolve the issue of class imbalance problem during training.	Detection speed is slow as compared to other single stage detectors.	Medium	Medium	High

In [30], a subcategory-aware CNN model is proposed for object detection. The model generates a feature pyramid of various scales with Region of Interest (RoI) pooling and a final convolutional layer used for subcategorization. One of the challenges in RDA is that a vehicle at a long- distance is seen as a small-scale object. Therefore, object detection using CNN is still a challenging problem because pooling damages small-scale objects. To minimise the effect of pooling, [27] proposed a RoI pooling method implemented for small-scale object detection.

Another RoI-pooling based method to detect vehicles on the road is proposed in [31]. The proposed system used global average pooling to avoid overfitting issues. In [32], author proposed a cascade-CNN based vehicle detection method. Their work combined two different CNNs. First, the network handles variant data on a small scale. The second CNN is designed for feature extraction, selection, and decision-making. After individual implementation, both CNN combined to achieve better performance. But their proposed network failed to detect extremely small vehicles and vehicles with a high inter-class variation. Ashraf et al.,[22] proposed Hybrid Vehicle Detection Network (HVD-Net) for traffic surveillince applications. They introduce Dense Connection Block (DCB) and Dense Spatial Pyramid Pooling (DSPP) block to handle class variation and small sized vehicle detection. DSPP becomes ineffective if the window size increases to a large number and increase complexti [33]. Moreover multi scale windows increase model complexity and impede end-to-end training. Table 1 critically analyses and briefly describes various vision-based detectors.

3. Proposed Method

A. Vehicle Detection

The proposed vehicle detection model integrates the strengths of the YOLO object detection system, which is known for its efficiency and accuracy in real-time object detection tasks. The core design rationale behind our approach is to optimize vehicle detection in dynamic environments, such as roads or highways, by leveraging the benefits of YOLO.

The chosen architecture divides the input video frame into a 13x13 grid, a decision made based on prior evidence that such a division provides a balance between granularity and computational efficiency [22]. Each cell in this grid is designed to predict bounding boxes and confidence scores for those boxes. By dividing the frame into grid cells, we allow the model to localize vehicle objects spatially. This granularity ensures that even when multiple vehicles are closely packed or overlap in a scene, the model can distinctly recognize and categorize each one [21].

If the vehicle centroid falls into a grid cell, then that grid is responsible for vehicle detection. Out of 13x13 grid cells, each grid cell estimates confidence scores and bounding boxes. The confidence score is pivotal as it provides a probabilistic measure of detection accuracy. The confidence score (i.e., $P_r(Vehicle) * IoU_{Predicted}^{Ground\_Truth}$) indicates whether or not the box contains a vehicle. The confidence score should be zero if there is no vehicle present. If the vehicle is present in the box, the model tries to use the confidence score to equal the Intersection over Union (IoU) between the ground truth and predicted Bounding Box (BB). Each BB predicts five components: the vehicle's centre (i.e., x and y coordinates), the confidence score, and the predicted vehicle's height and width. The SIH vehicle detection mechanism shown in Fig. 3.

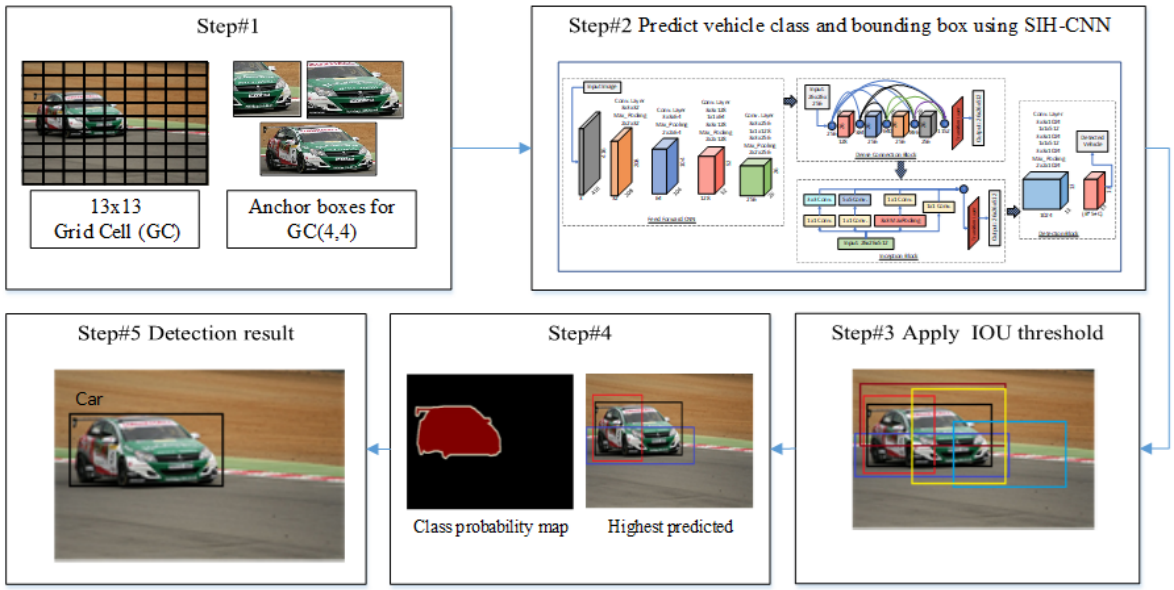


Fig. 3. Vehicle Detection Mechanism

B. Scale Invariant Hybrid Convolutional Neural Network (SIH-CNN)

The To address the problems of the baseline Darknet-19 backbone architecture, this study presents the SIH-CNN backbone architecture. SIH-CNN shown in Fig. 4 initially downsamples the convolutional structure to extract the features using multiple convolution and max-pooling units. On a 416x416 input image, the first unit performs Convolution_1 with 32 3x3 filters and a stride value of 1. Following Convolution_1, a max pooling layer was applied to

416x416x32 using a 2x2 filter with an S value of 2, yielding 208x208x32 feature maps. The entire CNN architecture used the same parameters for max pooling as used in the initial max pooling layer. Convolution_2 is performed in the second unit using 64 kernals of 3x3 size, and return feature maps of size of 104x104x64 after max-pooling_2. Three convolutional layers are used in the third unit: Convolution_3 (3x3), Convolution_4 (1x1), and Convolution_5 (filter size 3x3), with 128, 64, and 128 kernals, respectively.

After 3rd max pooling, we get 52x53x128 feature maps. The fourth unit utilized 3- Convolution layers (i.e., 256-Convolution_6 (filter size 3x3), 128-Convolution_7 (filter size 1x1), and 256-Convolution_8 (filter size 3x3)), and then 4th max pooling returns 26x26x256 feature maps.

After the fourth unit, we get more robust feature maps. To improve feature extraction, we now present the Dense Connection Block (DCB), which also mitigates the impact of gradient vanishing during in-network backpropagation. In the DC architecture, feature maps from Layer L-1 are combined with those from the present Layer L and the following Layer L+1. Too many DC-convolution layers in the CNN design slow down detection and add complexity to the model. Taking this into account, SIH-CNN uses DCB in the second-last convolutional block to derive the most informative semantic features. The suggested SIH-CNN design utilizes a dense connection module with four DC units, each of which is made up of a 1x1 and a 3x3 Conv-layer. Each DC unit incorporates a batch normalization (BN) layer before its 3x3 convolutional layer.

After DCB, SIH-CNN utilised inception block. To extract multi-scale features, the previous layer's feature maps are processed with three different dimension filters (i.e., 1x1, 3x3, and 5x5) and one max-pooling filter at the start. Once the multi-scale feature maps have been extracted, they are chained together to form a robust framework suitable for recognizing the smaller cars. When the CNN is designed to execute all of its convolutions on the same layer, the network expands in width but not depth. as well as enhance the capability of extracting characteristics across multiple scales. The final convolutional layer receives the feature images that were generated by the inception block. The overfitting and total number of factors are then decreased by switching from the flatten layer to the global pooling layer.

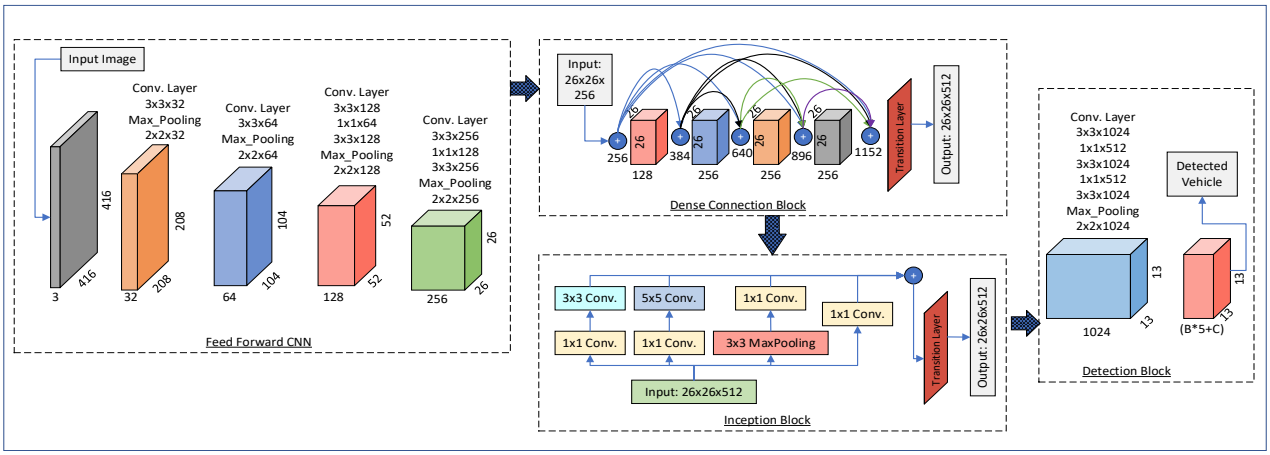


Fig. 4. Proposed SIH-CNN Backbone Architecture

4. Experiments

To assess the capability of suggested methodology all trials were performed on Intel(R) Core(TM) i7-7700K CPU, 4.50 GHz Max Turbo Frequency, and NVIDIA Titan X GPU with 12.00 GB memory.

A. Model Training

The SIH-CNN model employs Stochastic Gradient Descent (SGD) as its optimizer. SGD fine-tunes model weights by referencing the gradient of the loss function with respect to these weights. SGD is sensitive to its initial learning rate settings. Therefore, a learning rate of 0.0001 has been select. Typically, selecting for a smaller value is favored as it promotes stable convergence and mitigates the risk of bypassing the optimal solution point. SIH-CNN model utilizes the Leaky Rectified Linear Unit (Leaky ReLU). Functionally resembling the conventional ReLU, Leaky ReLU differentiates itself by holding a slight non-zero gradient for negative input values, ensuring no neuron falls into an inactive state during the learning process. And prevent the vanishing gradient problem and improve the model's ability to learn. The batch size selected for the SIH-CNN model training is 32. That determines the number of training examples processed in each iteration and affects both the speed and stability of the training process. A larger batch size lead to faster convergence but require more memory, while a smaller batch size lead to more stable convergence but slower.

B. Benchmark

In this research, the UA-DETRAC benchmark [35] was used to evaluate the proposed system in real-world traffic video sequences. The UA-DETRAC benchmark contains of a total of 1.21 million vehicle's boxes, covering four distinct vehicle categories, including cars, buses, vans, and others. In this research, 80% of benchmark is used for training and rest of the data used for evaluation. Using this benchmark, the proposed system is tested against a wide range of vehicle types and scales, providing a comprehensive evaluation of its performance. The benchmark provides ground truth data, allowing for the calculation of various performance metrics. Fig. 5 shows few samples from the UA-DETRAC benchmark, illustrating the variety of vehicle types and scales present in the dataset. By testing the proposed system against this benchmark, we can assess the ability of proposed CNN architecture to accurately detect vehicles in real-world scenarios, which is crucial for evaluating the system's effectiveness and potential in practical applications. Since the UA-DETRAC dataset covers real-world traffic data with different weather conditions and various vehicle classes, it can serve as a suitable proxy for evaluating the proposed SIH-CNN model's performance in a relevant context. The dataset's diversity allows for a comprehensive evaluation of the model's effectiveness in handling real-world scenarios, including vehicle detection in river dredging areas.

Fig. 5. UA-DETRAC benchmark samples

C. Evaluation Metrics

The mean Average Precision (mAP) is a commonly used metric to evaluate the performance of object detectors, including vehicle detectors. It measures the precision of the detection algorithm by calculating the average precision across all object classes and scales. This study utilized mAP to measure the performance of the proposed vehicle detector. The performance evaluation was conducted using the UA-DETRAC benchmark, which provided ground truth annotations for vehicle detection. Fig. 6 shows a sample image from the UA-DETRAC dataset, with examples of False Positive (FP), False Negative (FN), and True Positive (TP) cases that were used in the performance evaluation.

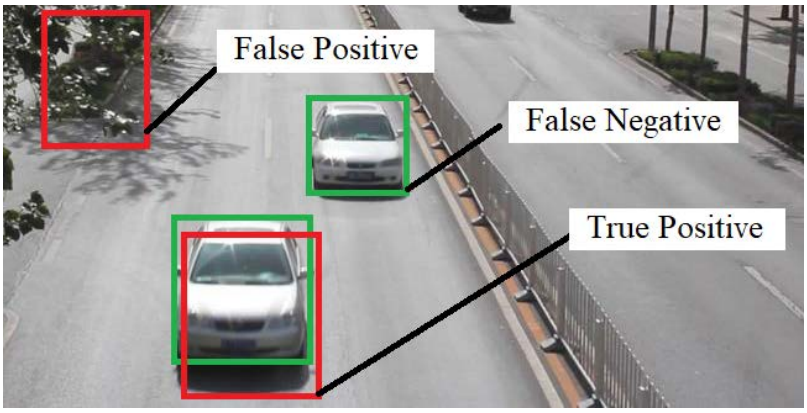


Fig. 6. UA-DETRAC detection sample

To calculate mAP, we used a set of performance evaluation matrices represented in equations(i)-(iv).

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}} \tag{1}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{2}$$

$$\text{Average precision (AP)}: = \int_0^1 \text{P(r)}\,\text{dr} = \frac{1}{11}\sum_{\text{rec}=0,0.1,0.2,\ldots,1} \text{Pinterp(rec)} \tag{3}$$

$$\text{mean Average precision (mAP)} = \frac{\sum_{i=1}^{k} \text{A}_{\text{Pi}}}{\text{k}} \tag{4}$$

$$\text{Frame Per Second (FPS)} = \frac{Number\ of\ frames}{\text{Duration in seconds}} \tag{5}$$

D. Performance Evaluation

The performance evaluation of the proposed SIH-CNN model was conducted by comparing its results with other well-known detection models, including [18-21].

Fig. 7-10 offer an in-depth comparative evaluation of the proposed SIH-CNN model against state-of-the-art methodologies. These sketches explain the Area Under the Curve (AUC) derived from the precision and recall metrics for detected vehicles across various classes, buses, cars, motorbikes, and trains. An AUC in a Precision-Recall (PR) curve is representative of a model's detection capability. The empirical results clearly showcase that the SIH-CNN model stands out, superior performance across all vehicle categories. It shows that SIH-CNN' as an exceptionally effective and precise model, real-world vehicle detection.

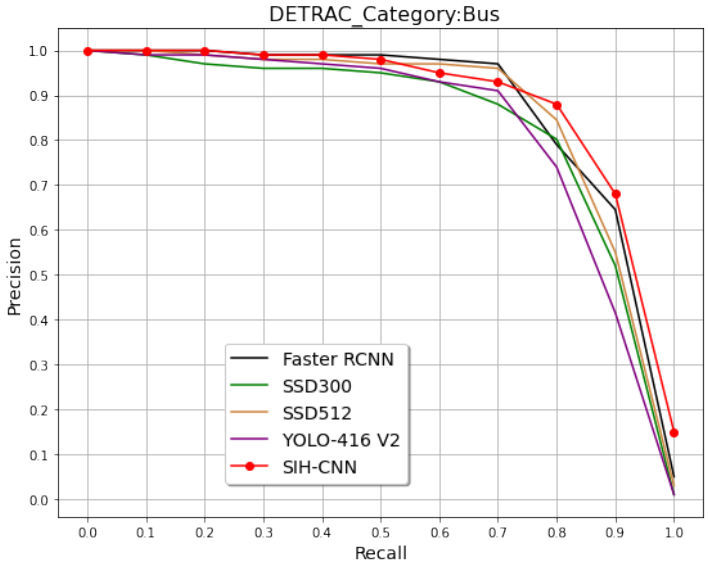


Fig. 7. AUC of Bus class

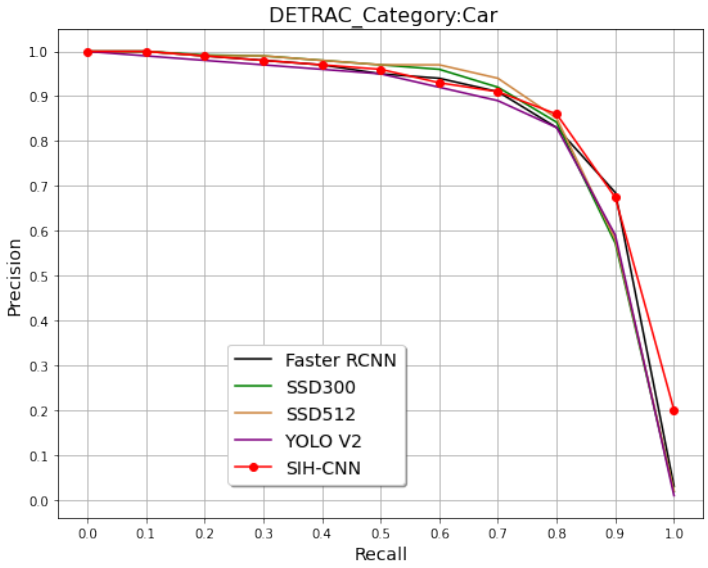


Fig. 8. AUC of Car class

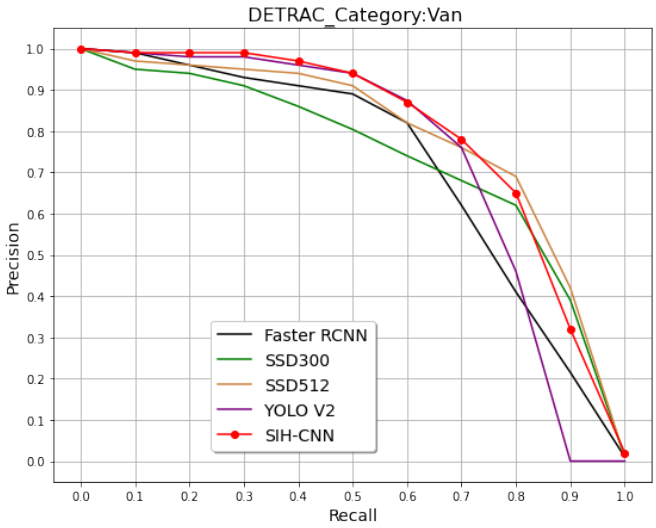


Fig. 9. AUC of Van class

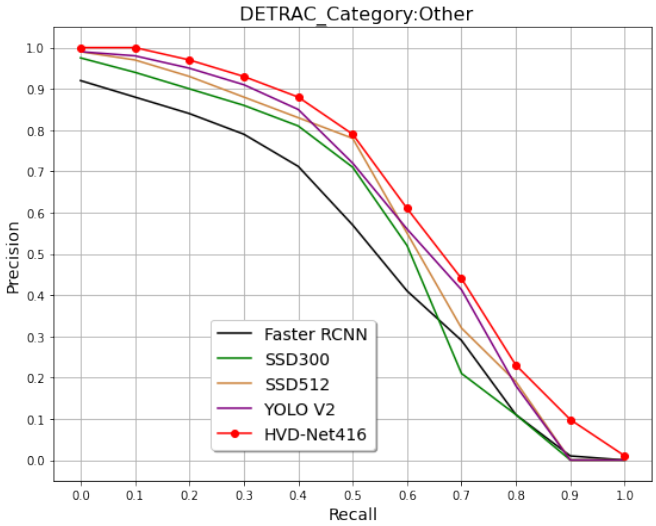


Fig. 10. AUC of Others class

Table 2. shows a broad performance evaluation of different vehicle detection models using the UA-DETRAC dataset, concentrating on four distinct vehicle classes: buses, cars, vans, and others. The proposed SIH-CNN model shows superior performance by attaining the highest mAP score of 77.76%. This model impressively scores the highest AP for cars (86.03%), vans (77.25%), and the category labeled as others (63.15%). While the AP score for buses, at 84.61%, is slightly lower than the Faster-RCNN model. But, the Faster-RCNN model achieves an mAP of 72.67%, SSD300 model records 73.08%, SSD512 touches 75.99%, and the YOLO-V2 model attains 73.82%.

Diving deeper into speed metrics, the Faster-RCNN model, built on the VGG-16 backbone, presents an mAP of 72.67% with a frame rate of 12.7 FPS. This processing speed underscores its high accuracy and may restrict its deployments in real-time surveillance applications. The SSD300 model, accomplishes an mAP of 73.08% and a commendable speed of 25.2 FPS. While its other variant, the SSD512, demonstrates a superior mAP of 75.99% but trails slightly in speed at 19.6 FPS. The YOLO-V2 model, underpinned by the DarkNet-19 architecture, observes itself with an FPS rate of 51.7, making it ideal for real-time surveillance, while maintaining an mAP of 73.82%. Evidently, the proposed SIH-CNN detector stands out, not only by achieving the high point mAP of 77.76% across the evaluated models but also by claiming a notable frame rate of 48.4 FPS. This highlights the model's capability to set the benchmark in both accuracy and speed in vehicular detection activities for surveillance applications.

The spider graph in Fig. 11 compares the average precision of all four vehicle classes (i.e., car, bus, van, and others) for different detectors. The graph shows that the proposed SIH-CNN model outperforms all other detectors in terms of average precision for all classes, with the largest area covered by the SIH-CNN curve. This indicates that the proposed model is more accurate and reliable in detecting different types of vehicles than other state-of-the-art detectors.

Table 2. Performance comparison

Vehicle Detection Framework	CNN Architecture	Input Size	Buses AP %	Cars AP %	Vans AP %	Others Catogery AP %	mAP %	FPS
Faster-RCNN [18]	VGG-16	600x600	85.49	84.4	70.49	50.29	72.67	12.7
SSD300 [20]	VGG-16	300x300	81.56	84.05	71.85	54.86	73.08	25.2
SSD512 [20]	VGG-16	512x512	84.32	84.46	76.64	58.55	75.99	19.6
YOLO-V2 Baseline [21]	DarkNet-19	416x416	80.86	82.63	72.22	59.57	73.82	51.7
Proposed Framework	SIH-CNN	416x416	84.61	86.03	77.25	63.15	77.76	48.4

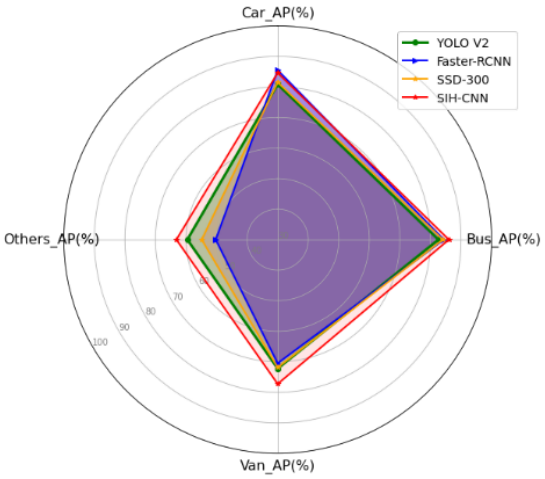


Fig. 11. Average precision of all vehicle classes

The qualitative results of vehicle detection on the UA-DETRAC benchmark are shown in Fig. 12. The proposed framework detects vehicles of different sizes with their class labels using traffic video frames.

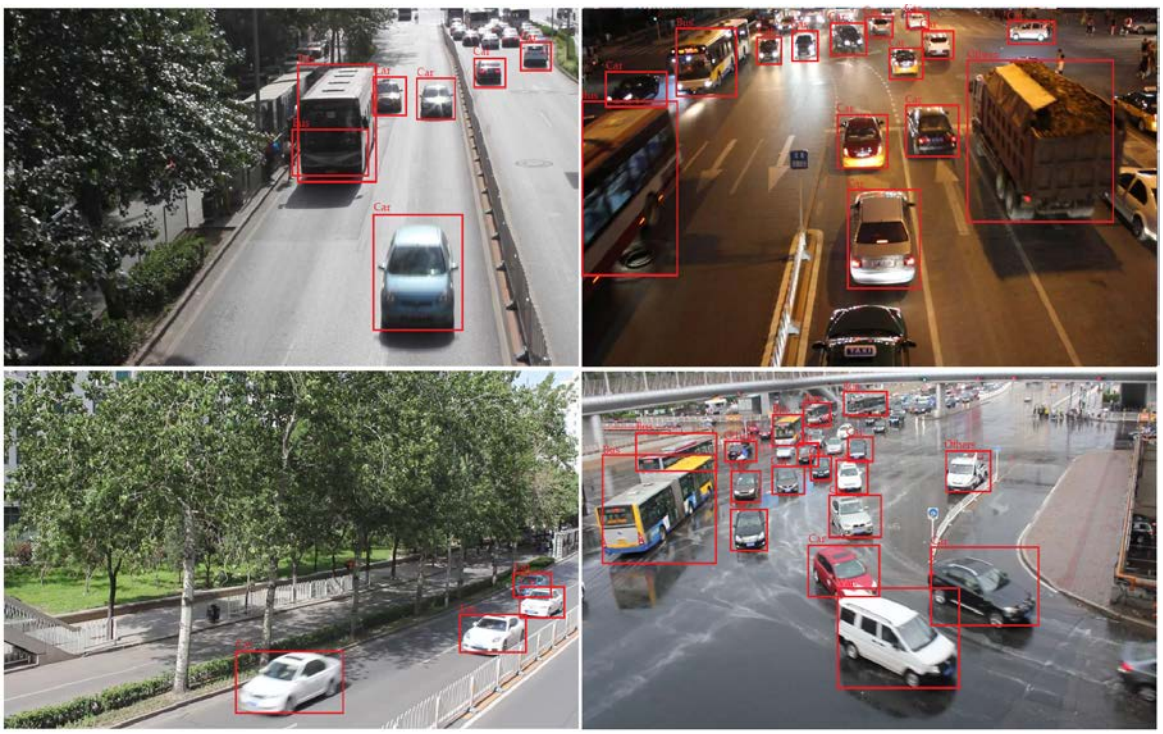


Fig. 12. Vehicle detection

5. Conclusion

The proposed SIH-CNN model has been utilized for real-time vehicle detection for surveillance applications like river dredging areas. The model combines multilevel and multiscale features that increase the feature extraction capabilities of the SIH-CNN architecture. And enable SIH-CNN to detect vehicles with scale changes while maintaining the original structures of vehicles. Real-world traffic surveillance benchmark is used to justify the strength of the proposed SIH-CNN model, and results proved that it outclass well-known vehicle detection models on real world data. In conclusion, the suggested SIH-CNN model can be used for real-time vehicle detection in river dredging areas surveillance applications, with potential applications in other similar scenarios. In future work, the same multilevel and multiscale features will be employed to design a vehicle license plate recognition procedure, that is the next phase of vehicle surveillance in river dredging areas.

References

[1] L. He, S. Wen, L. Wang, and F. Li, "Vehicle theft recognition from surveillance video based on spatiotemporal attention," *Applied Intelligence*, vol. 51, no. 4, pp. 2128–2143, Apr. 2021, doi: 10.1007/S10489-020-01933-8/TABLES/7.

[2] J. D. Trivedi, S. D. Mandalapu, and D. H. Dave, "Vision-based real-time vehicle detection and vehicle speed measurement using morphology and binary logical operation," *J Ind Inf Integr*, vol. 27, p. 100280, May 2022, doi: 10.1016/J.JII.2021.100280.

[3] P. J. Recky, "Total Solution For Smart Traffic and Toll Roads Management in Indonesia," *Devotion Journal of Community Service*, vol. 3, no. 2, pp. 149–157, Dec. 2021, doi: 10.36418/DEV.V3I2.119.

[4] Z. Wang, J. Huang, N. N. Xiong, X. Zhou, X. Lin, and T. L. Ward, "A Robust Vehicle Detection Scheme for Intelligent Traffic Surveillance Systems in Smart Cities," *IEEE Access*, vol. 8, pp. 139299–139312, 2020, doi: 10.1109/ACCESS.2020.3012995.

[5] J. S. Chou and C. H. Liu, "Automated Sensing System for Real-Time Recognition of Trucks in River Dredging Areas Using Computer Vision and Convolutional Deep Learning," *Sensors 2021, Vol. 21, Page 555*, vol. 21, no. 2, p. 555, Jan. 2021, doi: 10.3390/S21020555.

[6] H. A. Saad and E. H. Habib, "Assessment of Riverine Dredging Impact on Flooding in Low-Gradient Coastal Rivers Using a Hybrid 1D/2D Hydrodynamic Model," *Frontiers in Water*, vol. 3, p. 628829, Mar. 2021, doi: 10.3389/FRWA.2021.628829/BIBTEX.

[7] A. Xu, L. E. Yang, W. Yang, and H. Chen, "Water conservancy projects enhanced local resilience to floods and droughts over the past 300 years at the Erhai Lake basin, Southwest China," *Environmental Research Letters*, vol. 15, no. 12, p. 125009, Dec. 2020, doi: 10.1088/1748-9326/ABC588.

[8] D. Kusumaningrum, T. A. Hafsari, and L. Syam, "Sand and The City: The historical geography of sand mining in Jeneberang River and its relation to urban development in South Sulawesi," *ETNOSIA : Jurnal Etnografi Indonesia*, vol. 6, no. 2, pp. 200–216, Nov. 2021, doi: 10.31947/ETNOSIA.V6I2.17918.

[9] J. S. Chou and Y. C. Chiu, "Identifying critical risk factors and responses of river dredging projects for knowledge management within organisation," *J Flood Risk Manag*, vol. 14, no. 1, p. e12690, Mar. 2021, doi: 10.1111/JFR3.12690.

[10] Z. Yang and L. S. C. Pun-Cheng, "Vehicle detection in intelligent transportation systems and its applications under varying environments: A review," *Image Vis Comput*, vol. 69, pp. 143–154, Jan. 2018, doi: 10.1016/J.IMAVIS.2017.09.008.

[11] S. Sivaraman and M. M. Trivedi, "Looking at vehicles on the road: A survey of vision-based vehicle detection, tracking, and behavior analysis," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 4, pp. 1773–1795, Dec. 2013, doi: 10.1109/TITS.2013.2266661.

[12] A. ; Alarbi, Z. Albayrak, A. Forestiero, A. Alarbi, and Z. Albayrak, "Core Classifier Algorithm: A Hybrid Classification Algorithm Based on Class Core and Clustering," *Applied Sciences 2022, Vol. 12, Page 3524*, vol. 12, no. 7, p. 3524, Mar. 2022, doi: 10.3390/APP12073524.

[13] A. H. Ahmed, H. B. Alwan, and M. Çakmak, "Convolutional Neural Network-Based Lung Cancer Nodule Detection Based on Computer Tomography," *Lecture Notes in Networks and Systems*, vol. 572, pp. 89–102, 2023, doi: 10.1007/978-981-19-7615-5_8/COVER.

[14] K. W. Al-Mansoori and M. Cakmak, "Automatic Speech Recognition (ASR) System using convolutional and Recurrent neural Network Approach," *HORA 2022 - 4th International Congress on Human-Computer Interaction, Optimization and Robotic Applications, Proceedings*, 2022, doi: 10.1109/HORA55278.2022.9799877.

[15] H. C. Altunay and Z. Albayrak, "A hybrid CNN+LSTM-based intrusion detection system for industrial IoT networks," *Engineering Science and Technology, an International Journal*, vol. 38, p. 101322, Feb. 2023, doi: 10.1016/J.JESTCH.2022.101322.

[16] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation." pp. 580–587, 2014. Accessed: Jul. 29, 2023. [Online]. Available: http://arxiv.

[17] R. Girshick, "Fast R-CNN." pp. 1440–1448, 2015. Accessed: Jul. 29, 2023. [Online]. Available: https://github.com/rbgirshick/

[18] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *Adv Neural Inf Process Syst*, vol. 28, 2015, Accessed: Jul. 29, 2023. [Online]. Available: https://github.com/

[19] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection." pp. 779–788, 2016. Accessed: Jul. 29, 2023. [Online]. Available: http://pjreddie.com/yolo/

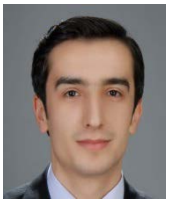
[20] W. Liu *et al.*, "SSD: Single shot multibox detector," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9905 LNCS, pp. 21–37, 2016, doi: 10.1007/978-3-319-46448-0_2/FIGURES/5.

[21] J. Redmon and A. Farhadi, "YOLO9000: Better, Faster, Stronger." pp. 7263–7271, 2017. Accessed: Jul. 29, 2023. [Online]. Available: http://pjreddie.com/yolo9000/

[22] M. H. Ashraf, F. Jabeen, H. Alghamdi, M. S. Zia, and M. S. Almutairi, "HVD-Net: A Hybrid Vehicle Detection Network for Vision-Based Vehicle Tracking and Speed Estimation," *Journal of King Saud University - Computer and Information Sciences*, vol. 35, no. 8, p. 101657, Sep. 2023, doi: 10.1016/J.JKSUCI.2023.101657.
[23] H. Alghamdi and T. Turki, "PDD-Net: Plant Disease Diagnoses Using Multilevel and Multiscale Convolutional Neural Network Features," *Agriculture 2023, Vol. 13, Page 1072*, vol. 13, no. 5, p. 1072, May 2023, doi: 10.3390/AGRICULTURE13051072.
[24] C. Kyrkou, "YOLOpeds: efficient real-time single-shot pedestrian detection for smart camera applications," *IET Computer Vision*, vol. 14, no. 7, pp. 417–425, Oct. 2020, doi: 10.1049/IET-CVI.2019.0897.
[25] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal Loss for Dense Object Detection." pp. 2980–2988, 2017.
[26] L. Wang, Y. Lu, H. Wang, Y. Zheng, H. Ye, and X. Xue, "Evolving boxes for fast vehicle detection," *Proc (IEEE Int Conf Multimed Expo)*, pp. 1135–1140, Aug. 2017, doi: 10.1109/ICME.2017.8019461.
[27] X. Hu *et al.*, "SINet: A Scale-Insensitive Convolutional Neural Network for Fast Vehicle Detection," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 3, pp. 1010–1019, Mar. 2019, doi: 10.1109/TITS.2018.2838132.
[28] F. Zhang, F. Yang, C. Li, and G. Yuan, "CMNet: A connect-and-merge convolutional neural network for fast vehicle detection in urban traffic surveillance," *IEEE Access*, vol. 7, pp. 72660–72671, 2019, doi: 10.1109/ACCESS.2019.2919103.
[29] L. Chen, F. Ye, Y. Ruan, H. Fan, and Q. Chen, "An algorithm for highway vehicle detection based on convolutional neural network," *EURASIP J Image Video Process*, vol. 2018, no. 1, pp. 1–7, Dec. 2018, doi: 10.1186/S13640-018-0350-2/TABLES/2.
[30] Y. Xiang, W. Choi, Y. Lin, and S. Savarese, "Subcategory-Aware convolutional neural networks for object proposals & detection," *Proceedings - 2017 IEEE Winter Conference on Applications of Computer Vision, WACV 2017*, pp. 924–933, May 2017, doi: 10.1109/WACV.2017.108.
[31] H. Haritha and S. K. Thangavel, "A modified deep learning architecture for vehicle detection in traffic monitoring system," *https://doi.org/10.1080/1206212X.2019.1662171*, vol. 43, no. 9, pp. 968–977, 2019, doi: 10.1080/1206212X.2019.1662171.
[32] X. Wu, X. Chen, and J. Zhou, "C-CNN: Cascaded convolutional neural network for small deformable and low contrast object localization," *Communications in Computer and Information Science*, vol. 771, pp. 14–24, 2017, doi: 10.1007/978-981-10-7299-4_2/FIGURES/8.
[33] Z. Xia and J. Kim, "Mixed spatial pyramid pooling for semantic segmentation," *Appl Soft Comput*, vol. 91, p. 106209, Jun. 2020, doi: 10.1016/J.ASOC.2020.106209.
[34] J. Dai, Y. Li, K. He, and J. Sun, "R-FCN: Object Detection via Region-based Fully Convolutional Networks," *Adv Neural Inf Process Syst*, vol. 29, 2016, Accessed: Jul. 29, 2023. [Online]. Available: https://github.com/daijifeng001/r-fcn.
[35] L. Wen *et al.*, "UA-DETRAC: A new benchmark and protocol for multi-object detection and tracking," *Computer Vision and Image Understanding*, vol. 193, p. 102907, Apr. 2020, doi: 10.1016/J.CVIU.2020.102907.

Authors' Profiles

Mohammed Abduljabbar Zaid received B.Sc. in Computer Engineering from Al Mamoun University College, Baghdad, Iraq in 2014; He received his MSc. degree in Computer Engineering in 2023 from Karabuk University, Karabuk, Turkey. His research interest includes Object Detection. His professional activities have been focused on Real-Time Object Detection System, Vehicle Counting Model.



Muhammet Çakmak received his MSc. and PhD degree in Computer Engineering, Karabuk University, Karabuk, Turkey. He is currently working as an Assistant Professor at Karabuk University Electrical and Electronic Engineering. His research interest includes Computer Networks, Cyber Security, and Deep Learning.

How to cite this paper: Mohammed Abduljabbar Zaid Al Bayati, Muhammet Çakmak, "Real-Time Vehicle Detection for Surveillance of River Dredging Areas Using Convolutional Neural Networks", International Journal of Image, Graphics and Signal Processing(IJIGSP), Vol.15, No.5, pp. 17-28, 2023. DOI:10.5815/ijigsp.2023.05.02