Published by :
http://www.ijert.org

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 13 Issue 07, July-2024

Traffic Management using YOLO and Convolutional Fizzy Neural Network

Authors: Lokesh Dhake, Ashutosh Talekar, Krishna Bembade

Abstract:
As urbanization accelerates, the surge in intercity travel has led to various traffic-related issues like congestion and an overwhelming variety of vehicles. Addressing these challenges necessitates robust road data collection. Thus, this paper introduces an advanced traffic-monitoring system integrating You Only Look Once (YOLO) and a Convolutional Fuzzy Neural Network (CFNN) for recording traffic volume and vehicle types on roads. Initially, YOLO detects vehicles and combines with a vehicle-counting technique to gauge traffic flow. Subsequently, two efficient models (CFNN and Vector-CFNN) along with a network mapping fusion approach are proposed for vehicle classification. In experimental trials, our method achieved a commendable 90.45% accuracy on the Beijing Institute of Technology dataset. On the GRAM-RTM dataset, the YOLO-CFNN and YOLO-VCFNN classification methods demonstrated superior mean average precision and F1 measure (F1) of 99%, surpassing other methods. Field tests on Taiwanese roads showcased not only high F1 scores for vehicle classification but also remarkable accuracy in vehicle counting using the proposed YOLO-CFNN and YOLO-VCFNN methods. Moreover, the system maintains a detection speed of over 30 frames per second on the AGX embedded platform, underscoring its suitability for real-time vehicle classification and counting in practical settings.

Keywords:

Traffic-monitoring system, fuzzy neural network, vehicle classification fusion deep learning, YOLO.

INTRODUCTION:

In Mumbai, approximately 700 new vehicles hit the roads each day, as reported in [1], while the total road length remains steady at around 2000 kilometers. This has led to significant congestion due to ongoing development projects. Traffic violations stand out as a primary cause of road accidents, resulting in numerous casualties and injuries annually. Preventing such tragedies necessitates strict adherence to traffic rules. The adoption of advanced technology in traffic monitoring, such as widespread camera installations across cities, has enabled the capture of violations effectively. Leveraging this infrastructure presents an opportunity to enhance traffic conditions further. Machine Learning and Artificial Intelligence (AI) emerge as vital tools for processing the extensive data gathered through CCTV cameras. AI, encompassing subsets like Machine Learning and Deep Learning, offers the capability for computers to learn autonomously without explicit programming. With the aim of assisting humans and executing tasks with heightened precision, computers are directed to perform specific functions.

The study of road traffic monitoring holds significant importance. By analyzing vehicle types and traffic patterns, current traffic conditions can be comprehended, enabling the provision of actionable insights to traffic management authorities. This data aids in making decisions that enhance people's quality of life. For instance, during holidays, insights into road traffic volume can suggest alternate routes to drivers, alleviating congestion in certain areas. Moreover, if specific roads are frequented by large trucks, roadside warnings can be installed to alert drivers, thereby reducing traffic accidents. Furthermore, identifying and tracking vehicles based on type and color can assist in law enforcement efforts. These applications all hinge on data collected by road monitoring systems for analysis. Consequently, researchers have explored various methods for vehicle detection and classification to gather information on passing vehicles.

Traditional methods for vehicle detection can be broadly categorized into two types: static-based and dynamic-based approaches.

Static-based methods rely on analyzing stationary characteristics of vehicles within images. For instance, Mohamed et al. employed Haar-like features to capture vehicle shapes and then fed these features into an artificial neural network for classification. Similarly, Wen et al. utilized Haar-like features to extract edge and structural features of vehicles, employing AdaBoost to filter essential features which were then classified using a Support Vector Machine (SVM). Sun et al. and David and Athira utilized Garbor filters to characterize vehicles and subsequently employed SVMs to determine vehicle presence in images. Wei et al. introduced a two-step approach wherein Haar-like features and AdaBoost were first used to identify regions of interest containing vehicles, followed by the application of Histogram of Oriented Gradients (HOG) and SVM for region verification. Yan et al. developed a system utilizing vehicle shadows to delineate vehicle boundaries, leveraging HOG for feature extraction, and employing AdaBoost and SVM classifiers for verification. Notably, in this method, when vehicles obstruct each other, they are treated as one vehicle due to connected shadows, which may reduce detection accuracy.

Regarding dynamic approaches, Seenouvong et al. proposed a system for vehicle detection and counting based on dynamic features. They utilized background subtraction to generate a difference map, allowing segmentation of the foreground image. Various morphological operations were then employed to identify moving objects, extract their outlines and bounding boxes, and ultimately count vehicles passing through specified areas.

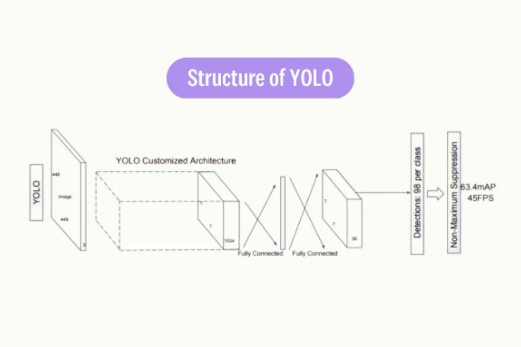
Some researchers have utilized Gaussian mixture models (GMMs) to model background scenes or adaptive backgrounds, aiming to address issues with background subtraction arising from gradual changes in brightness. However, both static and dynamic methods have limitations in addressing this problem. Traditional feature extraction methods often require manual design by experts based on their experience, leading to a complex process. Moreover, the features extracted are typically shallow and limited in their ability to effectively capture changes in vehicle characteristics.

Dynamic feature methods increase the complexity of subsequent image processing operations, particularly in scenarios with extensive background changes, and may yield suboptimal detection results. As deep learning techniques have advanced, conventional methods like these are gradually being replaced by deep learning approaches.

Model:

YOLOv3, or "You Only Look Once," stands out in the field of Object Detection, surpassing traditional algorithms in performance. Object detection involves two primary tasks: locating objects in an image and then classifying them. While previous methods like R-CNN showed improvements over traditional techniques, they faced limitations due to their complex pipelines, necessitating separate training for each component.

In contrast, YOLOv3 revolutionizes this approach by integrating both tasks into a single neural network. This unified architecture not only boosts accuracy but also streamlines the process. YOLOv3 offers notable advantages, including swift performance and the ability to detect multiple objects within a single image, setting it apart from its predecessors.



In an FCNN (Fully Convolutional Neural Network), the input image is divided into a grid of size S x S. Each cell within this grid is tasked with detecting any object that lies within its boundaries. When an object is detected, the cell predicts a bounding box and provides the confidence level indicating the likelihood of the object's presence. Each predicted bounding box includes five components:

- The (x, y) coordinates of the center of the box relative to the grid cell.
- The width (w) and height (h) of the predicted object.
- The confidence score, which is expressed as the Intersection over Union (IoU) between the predicted bounding box and the ground truth.

Additionally, each grid cell predicts the conditional class probability for the detected object.

Earlier versions of YOLO utilized the Darknet-19 architecture, which initially comprised 19 layers. Later, an additional 11 layers were incorporated for object detection . However, these versions faced challenges with accurately detecting small objects. Although concatenating feature maps was attempted to address this issue, it did not yield significant improvements.

The YOLOv3 architecture, on the other hand, consists of 106 convolutional layers and is a feature-learning based network. This architecture is a variant of the Darknet framework, which includes 53 layers pre-trained on the ImageNet dataset . An additional 53 layers are added to achieve state-of-the-art image detection. YOLOv3 can process images of any size without using pooling layers, thereby preserving fine details and minute features. Instead, convolutional layers with a stride of 2 are employed to downsample the feature map. The ResNet architecture, particularly the use of Residual Blocks, is crucial for enhancing both accuracy and speed.

Bounding Box Prediction

$$b_x = \sigma(t_x) + c_x$$

$$b_y = \sigma(t_y) + c_y$$

$$b_w = p_w e^{t_w}$$

$$b_h = p_h e^{t_h}$$

The bounding box coordinates (bx, by) and dimensions (bw, bh) represent the (x, y) coordinates, height, and width of the bounding box. The predicted values are denoted as tx, ty, tw, and th, while the top-left coordinates of the grid cell are represented by cx and cy. The anchor dimensions for the box are given by pw and ph.

The center coordinates, being offset values relative to the top-left of the grid cell, are passed through a sigmoid function. For example, if the top-left of the cell is at (8,8) and the center coordinate is (0.3, 0.8), then the center lies at (8.3, 8.8). Since the sigmoid function is applied, the center coordinates always range between 0 and 1.

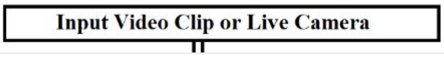
The height and width of the bounding box are calculated by applying a log-space transform to the output and then multiplying by the anchor dimensions. This output is then normalized to fall within the range of 0 to 1. For instance, if a feature map of size 13x13 is used and the values for bx and by are 0.4 and 0.6, the height and width of the bounding box are (13x0.4, 13x0.6).

The object score indicates the probability that an object is inside the bounding box. The value is 1 if the object's center lies within the grid cell and 0 if it lies at the grid corner. This score is also processed through a sigmoid function.

Class confidence represents the probability that the object belongs to a particular class. Initially, the softmax function was used for this purpose. However, because softmax assumes that objects belong to mutually exclusive classes, YOLOv3 now uses the sigmoid function for class confidence.

Flow chart:

Loss Function:



$$\lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right]$$
$$+ \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right]$$
$$+ \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left(C_i - \hat{C}_i \right)^2$$
$$+ \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{noobj}} \left(C_i - \hat{C}_i \right)^2$$
$$+ \sum_{i=0}^{S^2} \mathbb{1}_{i}^{\text{obj}} \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2$$

The loss function in versions prior to YOLOv3 is depicted in Fig. 4 [16]. The first term addresses the error in the offset of the bounding box location, specifically the (x, y) coordinates. The second term calculates the error related to the width and height of the bounding box. The third and fourth terms handle the object confidence error, while the fifth term assesses the class probability error [17]. All these errors are computed using the Sum of Squared Errors (SSE). However, in YOLOv3, the Cross-Entropy error replaces SSE, meaning that object confidence and class predictions are now made using logistic regression [18].

Literature Review:

In recent years, deep learning has gained widespread use across various fields, yielding impressive predictive results. Unlike traditional methods that rely on manually defined features, convolutional neural networks (CNNs) significantly enhance image recognition accuracy. Initially, Lecun et al. [14] introduced the LeNet model to address the challenge of recognizing handwritten digits in the banking sector. Krizhevsky et al. [15] advanced traditional CNNs with AlexNet by deepening the model architecture and incorporating the ReLU activation function and dropout layers to boost learning efficacy and prevent overfitting. Szegedy et al. [16] developed GoogLeNet, which employs multiple filters of varying sizes to extract more comprehensive feature information. Simonyan and Zisserman [17] proposed VGG-16 and VGG-19, models that replace large convolution kernels with successive small kernels to improve accuracy through increased model depth. He et al. [18] introduced the ResNet model, utilizing residual blocks to combat issues like gradient vanishing and convergence difficulties due to excessive network depth. Howard et al. [19] presented MobileNet, which uses depthwise separable convolutions to reduce redundant parameters while extracting fewer but more relevant features.

These advancements have significantly enhanced CNNs' feature description capabilities, extending their application to more complex tasks such as object detection. Several researchers [20]–[24] have leveraged region-based CNN (R-CNN) models to tackle vehicle detection. R-CNN uses a region proposal network (RPN) [25] to locate objects, which are then classified using a traditional CNN. RetinaNet [26] is a recent R-CNN model that employs a two-stage mechanism and a multilayer neural network for classification [27], [28]. However, this approach increases the number of parameters and reduces execution speed, making it unsuitable for real-time detection. To address this, one-stage methods like the YOLO (You Only Look Once) framework [29]–[31] and the single-shot multibox detector (SSD) [32] have been proposed. These methods offer real-time object detection but with lower classification accuracy compared to R-CNN methods [33], [34].

The existing object detection methods face several challenges: 1) Two-stage methods offer high classification accuracy but suffer from slow detection speeds due to a large number of parameters. 2) One-stage methods provide fast real-time detection but have lower accuracy than two-stage methods. 3) Expanding the number of object categories necessitates retraining the entire network, which is time-consuming and reduces scalability.

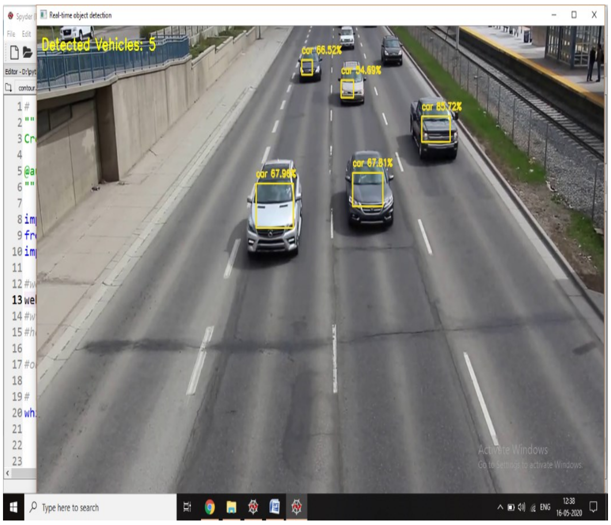
Recently, fuzzy neural networks (FNNs) [35]–[39] have combined the human-like fuzzy inference mechanism with the powerful learning capabilities of neural networks for tasks such as classification, control, and forecasting. Asim et al. [35] applied an adaptive network-based fuzzy inference system to classification problems, achieving higher accuracy than traditional neural networks. Lin et al. [36] utilized an interval type-2 FNN to predict tool flank wear, yielding excellent results. Other researchers have employed locally recurrent functional link FNNs [37] and Takagi-Sugeno-Kang-type FNNs [38], [39] for system identification and prediction, also with promising outcomes. In this study, an FNN was integrated into a deep learning network to reduce parameters and enhance classification accuracy. Conventional CNNs typically use pooling, global pooling [40], and channel pooling [41] for feature fusion. Global pooling methods, such as global average pooling (GAP) [42] and global max pooling (GMP) [43], sum spatial information to achieve robust feature fusion and prevent overfitting. Channel pooling methods, including channel average pooling (CAP) [44] and channel max pooling (CMP) [45], compute average or maximum values at corresponding positions in each channel of feature maps. However, these methods only compress features without learnable weights, leading to suboptimal classification results. This study proposes a new feature fusion method called network mapping to improve feature fusion effectiveness.

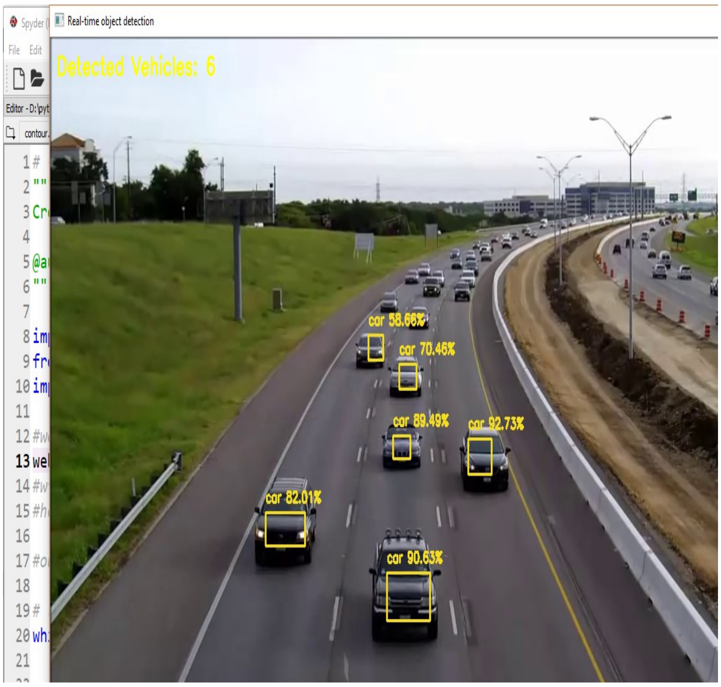
To develop an intelligent traffic-monitoring system with high execution speed, classification accuracy, and category extensibility, this study adopted a two-stage object detection approach. The proposed system, based on YOLO and a convolutional FNN (CFNN), collects real-time data on traffic volume and vehicle types. A novel modified YOLOv4-tiny (mYOLOv4-tiny) model is used for vehicle detection, combined with a vehicle counting method to calculate traffic flow. Additionally, two effective models (CFNN and Vector-CFNN) and a network mapping fusion method were proposed to enhance computational efficiency, classification accuracy, and category extensibility. The proposed model architecture has fewer parameters than other models, enabling real-time, high-accuracy vehicle classification with limited hardware resources and flexible category extension.

The contributions of this study are summarized as follows:

- Development of an intelligent traffic-monitoring system to record real-time traffic volume and vehicle types.
- Proposal of the mYOLOv4-tiny model for real-time object detection and improved detection efficiency.
- Implementation of two effective models (CFNN and Vector-CFNN) with a new network mapping fusion method to increase classification accuracy and significantly reduce model parameters.
- Enabling category extensions (e.g., vehicle type) by training only the classification model (CFNN), without retraining the object detection model (YOLO), thus saving substantial training time and improving category extension flexibility.
- Deployment of the proposed system on the NVIDIA AGX Xavier embedded platform for real-time vehicle tracking, counting, and classification on provincial highway 1 (T362) in Kaohsiung, Taiwan.

RESULTS:





The YOLOv3 model, trained on the COCO dataset, is utilized to detect 80 classes of objects, including motorcycles, bikes, cars, buses, trucks, and autorickshaws. The model underwent additional training to include autorickshaw detection using a dataset obtained from [19]. This training spanned 200 epochs and took approximately 48 hours on a machine equipped with an i5 processor and 8 GB of RAM, achieving a training loss of 0.0836.

Our focus is on detecting the following classes: ‘bicycle’, ‘car’, ‘motorcycle’, ‘bus’, ‘truck’, and ‘auto’.

We establish a virtual line aligned with the white line ahead of the zebra crossing, where vehicles are required to stop. If the traffic signal is red and a vehicle crosses this white line, a violation is recorded. Additionally, if a vehicle jumps a red light, an image of the vehicle must be captured.

CONCLUSION:

In this study, an intelligent traffic-monitoring system was proposed to calculate traffic flows and classify vehicle types. The major contributions of this study include:

- The development of a novel intelligent traffic-monitoring system combining a YOLOv4-tiny model with a counting method for traffic volume statistics and vehicle type classification.
- The design of the proposed CFNN and Vector-CFNN models by introducing the fusion method and FNN, which not only effectively reduce the number of network parameters but also enhance classification accuracy.

Experimental results showed that the proposed CFNN and Vector-CFNN models performed better than common deep learning models. On the BIT dataset, the network mapping fusion method improved recognition accuracy by 3.59%–5.92% compared to the pooling method. Compared to the PCN-Net model, the CFNN and Vector-CFNN models increased accuracy by 1.93% and reduced the number of parameters by 57.1%. On the GRAM-RTM dataset, the mAP and F1 scores of the proposed vehicle classification methods reached 99%, higher than those of other methods, and the proposed method was 1.65 times faster than traditional YOLOv4 based on FPS indicators. On the T362 vehicle type dataset, the network mapping fusion method's accuracy was 2.3%–5.36% higher than general pooling methods, and compared to the AlexNet model, the CFNN and Vector-CFNN models increased accuracy by 1.19% and 1.83%, respectively, while reducing parameters by 98.8%.

- The implementation of a network mapping fusion method that outperforms the commonly used pooling method, effectively integrating image features and improving classification accuracy.
- The proposed YOLO-CFNN and YOLO-VCFNN models demonstrated superior performance compared to current state-of-the-art object detection methods (Retinanet, SSD, YOLOv4, and YOLOv4-tiny), achieving high mAP rates, accurate counting accuracy, and real-time vehicle counting and classification capability (over 30 FPS).

In three actual road traffic scenarios, the YOLO-CFNN and YOLO-VCFNN methods achieved high F1 scores for vehicle classification and high accuracy for vehicle counting. In summary, the CFNN and Vector-CFNN models proposed in this study not only provide excellent vehicle classification but also have fewer parameters compared to other models, making them suitable for information analysis in environments with limited hardware performance.

Regarding the extensibility of the proposed models, many factors affecting the machining accuracy of machine tools in intelligent manufacturing, such as temperature and tool wear, have been identified. Therefore, developing accurate models for these factors is crucial. Future studies will apply the proposed CFNN and Vector-CFNN models and the network mapping fusion method for modeling in intelligent manufacturing.

REFERENCES:

[1] A. Mohamed, A. Issam, B. Mohamed, and B. Abdellatif, ‘‘Real-time detection of vehicles using the Haar-like features and artificial neuron networks,’’ Proc. Comput. Sci., vol. 73, pp. 24–31, Jan. 2015.

[2] X. Wen, L. Shao, W. Fang, and Y. Xue, ‘‘Efficient feature selection and classification for vehicle detection,’’ IEEE Trans. Circuits Syst. Video Technol., vol. 25, no. 3, pp. 508–517, Mar. 2015.

[3] Z. Sun, G. Bebis, and R. Miller, ‘‘On-road vehicle detection using Gabor filters and support vector machines,’’ in Proc. 14th Int. Conf. Digit. Signal Process. (DSP), Jul. 2002, pp. 1019–1022.

[4] H. David and T. A. Athira, ‘‘Improving the performance of vehicle detection and verification by log Gabor filter optimization,’’ in Proc. 4th Int. Conf. Adv. Comput. Commun., Aug. 2014, pp. 50–55.

[5] Y. Wei, Q. Tian, J. Guo, W. Huang, and J. Cao, ‘‘Multi-vehicle detection algorithm through combining Harr and HOG features,’’ Math. Comput. Simul., vol. 155, pp. 130–145, Jan. 2018.

[6] S. Bougharriou, F. Hamdaoui, and A. Mtibaa, ‘‘Linear SVM classifier based HOG car detection,’’ in Proc. 18th Int. Conf. Sci. Techn. Autom. Control Comput. Eng. (STA), Dec. 2017, pp. 241–245.

[7] G. Yan, M. Yu, Y. Yu, and L. Fan, ‘‘Real-time vehicle detection using histograms of oriented gradients and AdaBoost classification,’’ Optik, vol. 127, no. 19, pp. 7941–7951, 2016.

[8] N. Seenouvong, U. Watchareeruetai, C. Nuthong, K. Khongsomboon, and N. Ohnishi, ‘‘A computer vision based vehicle detection and counting system,’’ in Proc. 8th Int. Conf. Knowl. Smart Technol. (KST), Feb. 2016, pp. 224–227.

[9] P. K. Bhaskar and S.-P. Yong, ‘‘Image processing based vehicle detection and tracking method,’’ in Proc. Int. Conf. Comput. Inf. Sci. (ICCOINS), Jun. 2014, pp. 1–5.

[10] N. Seenouvong, U. Watchareeruetai, C. Nuthong, K. Khongsomboon, and N. Ohnishi, ‘‘Vehicle detection and classification system based on virtual detection zone,’’ in Proc. 13th Int. Joint Conf. Comput. Sci. Softw. Eng. (JCSSE), Jul. 2016, pp. 1–5.

[11] M. Anandhalli and V. P. Baligar, ‘‘Improvised approach using background subtraction for vehicle detection,’’ in Proc. IEEE Int. Advance Comput. Conf. (IACC), Jun. 2015, pp. 303–308.

[12] N. S. Sakpal and M. Sabnis, ‘‘Adaptive background subtraction in images,’’ in Proc. Int. Conf. Adv. Commun. Comput. Technol. (ICACCT), Feb. 2018, pp. 439–444.

[13] N. Shah, A. Pingale, V. Patel, and N. V. George, ‘‘An adaptive background subtraction scheme for video surveillance systems,’’ in Proc. IEEE Int. Symp. Signal Process. Inf. Technol. (ISSPIT), Dec. 2017, pp. 13–17.

[14] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, ‘‘Gradient-based learning applied to document recognition,’’ Proc. IEEE, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[15] A. Krizhevsky, I. Sutskever, and G. Hinton, ‘‘ImageNet classification with deep convolutional neural networks,’’ in Proc. 25th Int. Conf. Neural Inf. Process. Syst. (NIPS), vol. 1, Dec. 2012, pp. 1097–1105.

[16] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, ‘‘Going deeper with convolutions,’’ in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2015, pp. 1–9.

[17] K. Simonyan and A. Zisserman, ''Very deep convolutional networks for large-scale image recognition,'' 2014, arXiv:1409.1556.

[18] K. He, X. Zhang, S. Ren, and J. Sun, ''Deep residual learning for image recognition,'' in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 770–778.

[19] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, ''MobileNets: Efficient convolutional neural networks for mobile vision applications,'' 2017, arXiv:1704.04861.

[20] K. Shi, H. Bao, and N. Ma, ''Forward vehicle detection based on incremental learning and fast R-CNN,'' in Proc. 13th Int. Conf. Comput. Intell. Secur. (CIS), Dec. 2017, pp. 73–76.

[21] S.-C. Hsu, C.-L. Huang, and C.-H. Chuang, ''Vehicle detection using simplified fast R-CNN,'' in Proc. Int. Workshop Adv. Image Technol. (IWAIT), Jan. 2018, pp. 1–3.

[22] S. Rujikietgumjorn and N. Watcharapinchai, ''Vehicle detection with subclass training using R-CNN for the UA-DETRAC benchmark,'' in Proc. 14th IEEE Int. Conf. Adv. Video Signal Based Surveill. (AVSS), Aug. 2017, pp. 1–5.

[23] W. Zhang, Y. Zheng, Q. Gao, and Z. Mi, ''Part-aware region proposal for vehicle detection in high occlusion environment,'' IEEE Access, vol. 7, pp. 100383–100393, 2019.

[24] L. Wang, Y. Lu, H. Wang, Y. Zheng, H. Ye, and X. Xue, ''Evolving boxes for fast vehicle detection,'' in Proc. IEEE Int. Conf. Multimedia Expo. (ICME), Jul. 2017, pp. 1135–1140.

[25] S. Ren, K. He, R. Girshick, and J. Sun, ''Faster R-CNN: Towards realtime object detection with region proposal networks,'' IEEE Trans. Pattern Anal. Mach. Intell., vol. 39, no. 6, pp. 1137–1149, Jun. 2017.

[26] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, ''Focal loss for dense object detection,'' IEEE Trans. Pattern Anal. Mach. Intell., vol. 42, no. 2, pp. 318–327, Feb. 2020.

[27] N. A. Al-Sammarraie, Y. M. H. Al-Mayali, and Y. A. Baker El-Ebiary, ''Classification and diagnosis using back propagation artificial neural networks (ANN),'' in Proc. Int. Conf. Smart Comput. Electron. Enterprise (ICSCEE), Jul. 2018, pp. 1–5.

[28] O. I. Abiodun, A. Jantan, A. E. Omolara, K. V. Dada, N. A. Mohamed,and H. Arshad, ''State-of-the-art in artificial neural network applications: A survey,'' Heliyon, vol. 4, no. 11, Nov. 2018, Art. no. e00938.

[29] J. Redmon and A. Farhadi, ''YOLOv3: An incremental improvement,''2018, arXiv:1804.02767.

[30] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, ''YOLOv4: Optimalspeed and accuracy of object detection,'' 2020, arXiv:2004.10934.

[31] Z. Jiang, L. Zhao, S. Li, and Y. Jia, ''Real-time object detection methodbased on improved YOLOv4-tiny,'' 2020, arXiv:2011.04244.

[32] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Fu, and A. C. Berg,''SSD: Single shot multibox detector,'' in Proc. Eur. Conf. Comput. Vis.,Amsterdam, The Netherlands, Oct. 2016, pp. 21–37.

[33] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer,Z. Wojna, Y. Song, S. Guadarrama, and K. Murphy, ''Speed/Accuracy trade-offs for modern convolutional object detectors,'' in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 3296–3297.

[34] P. Soviany and R. T. Ionescu, ''Optimizing the trade-off between singlestage and two-stage deep object detectors using image difficulty prediction,'' in Proc. 20th Int. Symp. Symbolic Numeric Algorithms Sci. Comput.(SYNASC), Sep. 2018, pp. 209–214.

[35] Y. Asim, B. Raza, A. K. Malik, A. R. Shahid, M. Faheem, and Y. J. Kumar,''A hybrid adaptive neuro-fuzzy inference system (ANFIS) approach for professional bloggers classification,'' in Proc. 22nd Int. Multitopic Conf. (INMIC), Nov. 2019, pp. 1–6.

[36] C.-J. Lin, J.-Y. Jhang, S.-H. Chen, and K.-Y. Young, ''Using an interval type-2 fuzzy neural network and tool chips for flank wear prediction,'' IEEE Access, vol. 8, pp. 122626–122640, 2020.

[37] D. K. Bebarta, R. Bisoi, and P. K. Dash, ''Locally recurrent functional link fuzzy neural network and unscented H-infinity filter for shortterm prediction of load time series in energy markets,'' in Proc. IEEE Power, Commun. Inf. Technol. Conf. (PCITC), Oct. 2015, pp. 663–670.

[38] J.-W. Yeh and S.-F. Su, ''Efficient approach for RLS type learning in TSK neural fuzzy systems,'' IEEE Trans. Cybern., vol. 47, no. 9, pp. 2343–2352, Sep. 2017.

[39] C.-J. Lin, C.-H. Lin, and J.-Y. Jhang, ''Dynamic system identification and prediction using a self-evolving Takagi–Sugeno–Kang-type fuzzy CMAC network,'' Electronics, vol. 9, no. 4, p. 631, Apr. 2020.

[40] M. Lin, Q. Chen, and S. Yan, ''Network in network,'' 2013,arXiv:1312.4400.

[41] Z. Ma, D. Chang, J. Xie, Y. Ding, S. Wen, X. Li, Z. Si, and J. Guo, ''Finegrained vehicle classification with channel max pooling modified CNNs,'' IEEE Trans. Veh. Technol., vol. 68, no. 4, pp. 3224–3233, Apr. 2019.

[42] V. Christlein, L. Spranger, M. Seuret, A. Nicolaou, P. Kral, and A. Maier,''Deep generalized max pooling,'' in Proc. Int. Conf. Document Anal. Recognit. (ICDAR), Sep. 2019, pp. 1090–1096.

[43] Z. Li, S.-H. Wang, R.-R. Fan, G. Cao, Y.-D. Zhang, and T. Guo, ''Teeth category classification via seven-layer deep convolutional neural network with max pooling and global average pooling,'' Int. J. Imag. Syst. Technol., vol. 29, no. 4, pp. 577–583, May 2019.

[44] Z. Gao, Y. Li, Y. Yang, N. Dong, X. Yang, and C. Grebogi, ''A coincidence-filtering-based approach for CNNs in EEG-based recognition,'' IEEE Trans. Ind. Informat., vol. 16, no. 11, pp. 7159–7167, Nov. 2020.

[45] L. Cheng, D. Chang, J. Xie, R. Ma, C. Wu, and Z. Ma, ''Channel max pooling for image classification,'' in Intelligence Science and Big Data Engineering. Visual Data Engineering, Z. Cui, J. Pan, S. Zhang, L. Xiao, and J. Yang, Eds. Cham, Switzerland: Cham, Switzerland: Springer, 2019, pp. 273–284.

[46] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, ''Simple online and realtime tracking,'' in Proc. IEEE Int. Conf. Image Process. (ICIP), Sep. 2016, p. 346.

[47] J. Ou and Y. Li, ''Vector-kernel convolutional neural networks,'' Neurocomputing, vol. 330, pp. 253–258, Feb. 2019.

[48] N. Talpur, M. N. M. Salleh, and K. Hussain, ''An investigation of membership functions on performance of ANFIS for solving classification problems,'' IOP Conf. Ser., Mater. Sci. Eng., vol. 226, Aug. 2017, Art. no. 012103.

[49] Z. Dong, Y. Wu, M. Pei, and Y. Jia, ''Vehicle type classification using a semisupervised convolutional neural network,'' IEEE Trans. Intell. Transp. Syst., vol. 16, no. 4, pp. 2247–2256, Aug. 2015.

[50] F. C. Soon, H. Y. Khaw, J. H. Chuah, and J. Kanesan, ''Semisupervised PCA convolutional network for vehicle type classification,'' IEEE Trans. Veh. Technol., vol. 69, no. 8, pp. 8267–8277, Aug. 2020.

[51] R. Guerrero-Gómez-Olmedo, R. J. López-Sastre, S. Maldonado-Bascón, and A. Fernández-Caballero, ‘‘Vehicle tracking by simultaneous detection and viewpoint estimation,’’ in Natural and Artificial Computation in Engineering and Medical Applications, J. M. F. Vicente, J. R. Sánchez, F. de la Paz López, F. J. T. Moreo, Eds. Berlin, Germany: Springer, 2013, pp. 306–316.