ISSN: 0974-8571 Vol.13 No. 2 December, 2021

International Journal of Computational Intelligence in Control

CNN based Single-level and Multi-level Vehicle Classification Framework in Intelligent Transportation System

M. Swarnamugi [1]

[1] Research Scholar, VTU-RC , Department of MCA, CMR Institute of Technology
Assistant Professor, Department of MCA, Jyoti Nivas College, Bengaluru.
swathidevan@gmail.com

Dr. R. Chinnaiyan [2]

[2] Professor & Head, Department of Master of Computer Applications,
AMC Engineering College, Bangalore - 560083
vijayachinns@gmail.com , chinnaiyan.r@amceducation.in

Date of Submission: 21st October 2021 Revised: 11th November 2021 Accepted: 9th December 2021

How to Cite: **M. Swarnamugi & R. Chinnaiyan. (2021). CNN based Single-level and Multi-level Vehicle Classification Framework in Intelligent Transportation System. International Journal of Computational Intelligence in Control 13(2).**

Abstract

With advancement in image processing and computer vision, many machine learning based vehicle classification systems have been proposed in the literature. In traditional machine learning methodology, handcrafted feature extraction methods are used to extract important features to train the machine learning classifier to perform object level classification. This methodology works well in a controlled environment. However, with limited hand crafted features, this methodology is not suitable for generalization for vehicle classification algorithms in an unstructured environment. To overcome this limitation, we propose a new framework for vehicle classification for an unstructured environment using Convolutional Neural Network. The result and analysis of the proposed framework exemplify multi-level vehicle classification performs better than single-level classification.

Keywords: Intelligent transportation system, Vehicle classification, Convolutional Neural Network

1. Introduction

With the rapid growth in the number of vehicles in developing countries, the road traffic violation is increasing day by day. This leads to frequent traffic congestion, traffic accidents and potential risk for the safety of commuters, pedestrians and for the vehicles also. An effective traffic monitoring system will be a solution to overcome these issues. One of the ideal functionality of road traffic monitoring is vehicle classification. It is an integral part of the Intelligent Transportation System (ITS) for the effective transportation planning, control, and for the development of driverless or autonomous vehicles. Especially, in an unstructured road environment where the traffic rules are lightly followed, vehicle classification is an essential task to track traffic flow control, signal jumping and over speed. It also helps to identify and plan for new road pavements depending on the quantity of types of vehicles passing, predict future transportation needs and improve road safety.

Numerous vehicle classification methods have been proposed in the literature. The older type of vehicle classification is an in-roadway-based system which uses sensors such as loop induction sensors [1], vibration sensors [2], magnetic sensors and piezoelectric sensors [3]. These sensors have been deployed on the road pavements to collect information about the vehicles such as length of the vehicle and axle count to determine the type of vehicle. This methodology can accurately classify the vehicles because the sensors installed can have close contact with the moving vehicle. However, this methodology is rarely used nowadays due to undesired weather conditions which affects the sensors and also due to its high cost maintenance of breakages in the road pavement. With advancement in image processing and computer vision, many machine learning based vehicle classification systems have been proposed [4-6] in the literature. In traditional machine learning methodology, handcrafted feature extraction methods are used to extract important features to train the machine learning classifier to perform object level classification.

This methodology works well in a controlled environment. However, with limited hand crafted features, this methodology is not suitable for generalization where prior knowledge is essential.
For an unstructured environment where less adherence to traffic practices are followed, use of machine learning model for vehicle classification perform inappropriately and may result in serious flaws. To address this limitation, we propose a new framework for vehicle classification for an unstructured environment using Convolutional Neural Network. The contribution of this work include,

(i) We proposed a fine-tuned ResNet50 convolutional neural network model to improve the robustness of vehicle classification in an unstructured environment.
(ii) Two different tree hierarchies are proposed namely level 2 and level 3 for multi-level classification
(iii) Comparison of single-level and two different hierarchies of multi-level classifiers are performed and its results are discussed.

The rest of the paper is organized as follows. In Section II, the existing vehicle classification models based on CNN are described. In Section III, the deep learning methodology is explained. In Section IV the proposed approach namely single-level and multi-level classification framework is explained. In Section V, dataset formation and evaluation metric is explained. In Section VI, the experimental result, discussion and comparison is explained. Section VII concludes the proposed work.

2. Related Study

The vehicle classification is an important task in traffic monitoring for the Intelligent Transportation System. Many different methodologies have been proposed in the literature for vehicle classification. The prominent one being convolutional neural networks which demonstrated better accuracy for vision based image classification tasks [7]. In this section we will review the state of the art models and architectures proposed for vehicle classification using CNN. Dong et al. [8] have proposed a vehicle classification system. The presented model has been a semi – supervised model consisting of fully connected layer and softmax layer to classify real time vehicle data. The model has been tested on BIT-vehicle dataset on two different modes: the day and night with an accuracy of 96.01% and 89.6%. Maria et al. [9] have proposed a vehicle classification system using faster R-CNN and obtained an accuracy of 93% on their self-constructed dataset. In another work, Wang et al. [10] have proposed faster R-CNN based vehicle classification for real-time traffic monitoring systems. Their image dataset consists of more than 60, 000 images. Their model obtained an accuracy of 80.65%. Cao et al. [11] have presented a CNN classifier to work with incontinent road scenes. Their presented network consists of two parts: an end-to-end CNN architecture for coarse-to-live or top-down recognition task and vehicle classification framework.
Their work obtained an accuracy of 95.3% on the CompCars dataset. Jiang [12] presented a weakly – supervised CNN for vehicle classification with training relying on image-level labels. Their presented work has been tested on a self – constructed dataset captured from a traffic surveillance camera and the model obtained an accuracy of 98.28%. Valev et al. [13] have proposed a deep learning architecture for fine-grained vehicle classification using traditional data augmentation techniques such as flip, rotation, and blur. The model has been trained and tested on cars-196 dataset and achieved a classification accuracy of 94.6%. Chauhen et al. [14] have proposed an embedded CNN based vehicle classification and counting of vehicles in real-time Indian road traffic environment. They fine-tuned the YOLO CNN model and obtained an accuracy of 75 mAP. Jo et al. [15] have presented a CNN model based on pre-trained GoogleNet for vehicle classification and their proposed classifier has achieved an accuracy of 98.3% on ILSVRC – 2012 dataset. Karungaru et al. [16] have presented a vehicle type classification using fine-tuned AlexNet with SVM. A spatial pyramid pooling has been added to the network to solve the problem of image distortion caused during image resizing. Their presented model has been trained and tested on a BIT-Vehicle dataset.
Butt [17] have presented a CNN based vehicle classification using ResNet152 for ITS. The model has been trained and tested on a self-constructed dataset with more than 10, 000 images categorized into six classes. The model obtained an accuracy of 99.68%.

3. Methodology

The CNNs adopts a supervised learning strategy with a feed forward network for large scale object classification. CNNs are proven to considerably increase the performance of the classifier when applied to real world applications. Traditional methods for image classification include ML classifiers where the feature extractions are handcrafted. Compared to traditional ML classifiers, CNNs can automatically extract the learnable parameters from the input data to perform classification [18]. The architecture of CNN comprises three hierarchical layers, (i) Convolution layers, (ii) Pooling, and (iii) Fully connected layers.

3.1 Convolution Layer

The first and foremost layer in CNN is the convolutional layer. It comprises a set of learnable filters (kernels). The size of these filters are smaller than the input image and it slides over the input image during the forward pass and generates a two dimensional feature map. This feature map carries detected features in an input image such as edges, color, gradient and orientation etc. The feature map of the convolutional layer is calculated as [19],

$$k * I_{x,y} = \sum_{i=1}^{n} \sum_{j=1}^{n} k_{ij}. I_{x+i-1,\, y+j-1} + \beta \qquad (1)$$

Let I be an image with dimension $w \times h \times d$. $I_{x,y}$ refers to a position of a pixel on an image I. $k * I_{x,y}$ is a convolution operation which will modify the image I by applying a filter, and the filter is expressed as a matrix k of size $n \times n$. The beta is represented as β. The output produced by convolutional operation applied to an image is regarded as feature map and it has all the important features of an image I.

3.2 Pooling Layer

Pooling layer is used for down sampling the image size. It is applied to reduce the spatial size of the convolved feature to minimize computation and at the same time to retain an important feature of an input image which helps in controlling overfitting. Like the convolution layer, the pooling layer operation applies a filter across the input image but the difference is that they do not have any weights. Instead, the kernel applies an aggregation function within the receptive field generating an output array. Let the size of the incoming image to the pooling layer is $w \times h$. When pooling operation is performed the image size is reduced to,

$$\frac{w-k}{q+1} \times \frac{h-k}{q+1} \qquad (2)$$

Where, $w \times h$ is the width and height of the image, k is the kernel, q is a scalar value which dominates the number of pixels during the movement of kernel across the image [19]

There are different types of pooling operations generally used. They are max pooling and average pooling. With max pooling, as the kernel moves across the image, it returns the pixel with the maximum value. Whereas, with the average pooling, it returns the average of all the values within the receptive field covered by the kernel.

3.3 Fully connected layer

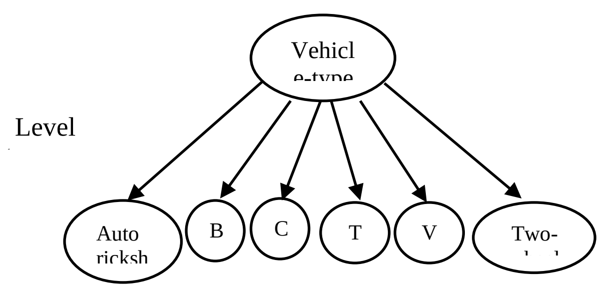
The last layer of CNN is neurons of fully connected layer where each neuron in the output layer is connected directly to a neuron in the previous layer to reduce the feature dimension. Each of these interconnections have a weight associated with the connection. The feed forward network initially starts with the random weight in the range of 0 to 1 and will be optimally turned after a few epochs. The feature maps generated from the final pooling layer of the CNN are converted into vectors and are passed to the fully connected layer. A back propagation is applied to every epoch in the fully connected layer. After a series of epochs, the model is able to classify the inputs using softmax activation function producing a probability of the classification correctness ranging from 0 to 1. To overcome the overfitting, a dropout operation is used to set the output of jth neuron with a probability. The output of a fully connected layer is a vector.

3.4 Data Augmentation

Data augmentation is a technique to alleviate overfitting from networks by artificially increasing the dataset through label-preserving transformation methods [20]. To increase the diversity of our dataset, we applied image transformation techniques during the training process. We employ techniques such as (i) Gaussian blur with a 5x5 filter that removes the high frequency noisy pixels while preserving the low frequency pixels. (ii) Rotation – we applied a ten degree rotation on the original dataset images to populate a varied view (iii) Horizontal flip – we employed a horizontal flip with a probability of 10% to increase the diversity of the images in the dataset for training the proposed CNN. (iv) Gaussian noise – we employed gaussian noise to make the training process more robust and to minimize the error.

4. Proposed Framework and its approaches

We use two different approaches namely single level classification (SLC) and multi-level classification (MLC). Single level is a flat hierarchy with six classification classes. SLC is assumed as a tree with one node and its level is one. Given an input image to this node, it acts as a classifier and gives probabilistic values for all class. A single - level classifier using CNN is depicted in Figure, we train a CNN classifier that predicts type of vehicle for labelling new instances such as auto-rickshaw, bus, car, truck, two-wheelers and van. Here, all classes are classified at the same level and these classes are independent of each other. Multi – level classification approach works on the hierarchy rule. In MLC, the classifier traverse the hierarchy from the root to the predicted leaf. This creates a connection between the nodes from the top level hierarchy to the bottom level leaf nodes. Furthermore, the hierarchy forms a tree with height greater than one and the classification nodes are the leaves of the hierarchy tree. Multi – level is a three level hierarchy with level 1, level 2 and level 3. When an image is passed as an input, the MLC approach estimate the probability of image belonging to a particular class from each root to leaf node through the hierarchy tree.



139

Fig. 1: Single Level Classifier

4.1 Approach I – Single-level classification

In single level classification, we have employed ResNet50 trained on ImageNet as the pretrained network for better classification accuracy. ResNet (Residual Network) is one of the most efficient and robust CNN architectures proposed by He et al. [21] and is best suitable to model classification and object recognition tasks. Vanishing gradient is a major problem in deep neural architectures. It states, when the network is deeper, the value of gradient shrinks to zero. As a result, the weights fail to update and therefore learning cannot happen. This makes the model perform poorly and due to which the model finds it difficult to converge. ResNet architecture handles this issue by introducing a novel skip connection where the input from the previous layer is accumulated to output of the next layer. According to the literature, fine tuning of a pretrained CNN network can increase the classification accuracy in the respective domain [22, 23]. The motivation behind the use of pretrained ResNet50 is to employ a transfer learning approach in single level classification. Transfer learning refers to a CNN model trained for one domain can be used to solve similar problems in related domains by learning new weights. This approach is best applicable when there is a lower amount of data insufficient for training from the scratch. Fig illustrate the transfer learning process.

In this section, a fine-tuned ResNet50 CNN for single level vehicle classification is explained. The initial layers of the ResNet50 are frozen as they learn more about simple features like edges and lines and these are common in all the objects. To perform transfer learning, the last fully connected layer, average pool, 1000-d fc, softmax is removed from the network, which was pre-trained to perform the classification of 1000 natural categories of ImageNet data. This layer is replaced with a classification block with three more layers: A fully connected layer with weight factor ‘’ as twenty and bias ‘β’ factor as twenty, the second layer is softmax layer and third layer is the classification layer that classifies six different vehicles in an unstructured Indian road scene. Fig. 1 depicts the architecture of the fine-tuned CNN model for single level classifier. The input to the network layer is 224 X 224 pixels RGB images. The model uses 64 convolution kernels of 7 X 7 size with the stride of 2 in the first layer and followed by 3 X 3 max pooling with the stride of 2.

In ResNet50 architecture, from layer two to layer five, there are a block of identical layers having several kernels to 128, 256, 1024 and 2048, followed by a fully connected layer. The second convolution layer conv2_x has 64 kernels of 1 X 1 filter size followed by 3 X 3, 64 kernels and at as 1 X 1, 256 kernels. These layers are repeated a total of three, having 9 layers in the second convolution. Similarly, convolution layer conv3_x is repeated four times having 12 layers, conv4_x is repeated six times having a total of 18 layers and conv5_x is repeated 3 times having 9 layers. The total number of layers in ResNet is thus 50 layers. The ResNet model enables gradients to flow using skip connections. It uses shortcuts namely identity shortcut and projection shortcut.

4.2 Approach II – Multi-level classification

Multi – level classification approach works on the hierarchy rule. In MLC, the classifier traverse the hierarchy from the root to the predicted leaf. This creates a connection between the nodes from the top level hierarchy to the bottom level leaf nodes. Furthermore, the hierarchy forms a tree with height greater than one and the classification nodes are the leaves of the hierarchy tree. Multi – level is a three level hierarchy with level 1, level 2 and level 3. When an image is passed as an input, the MLC approach estimate the probability of image belonging to a particular class from each root to leaf node through the hierarchy tree. This represents is-a relationship between the nodes. Two different MLC trees are constructed for experimentation purpose. Fig depicts a level 2 multi - level classifier hierarchy with tree height 2 and fig depicts a level 3 multi - level classifier hierarchy with tree height 3. In both the levels, each node apply one classification to the given input image. This leverage the hierarchical tree to be viewed as a divide and conquer problem where each sub problems are solved in multiple steps to improve the classification performance.

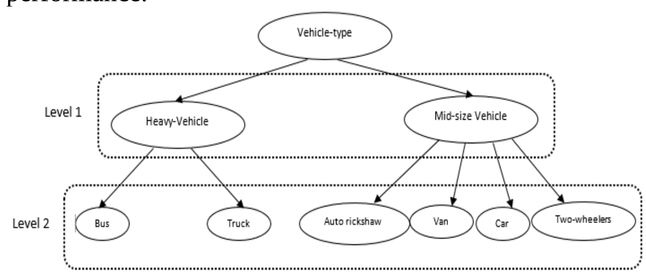


Fig. 2: Level 2 Hierarchy: Multi- Level Classifier

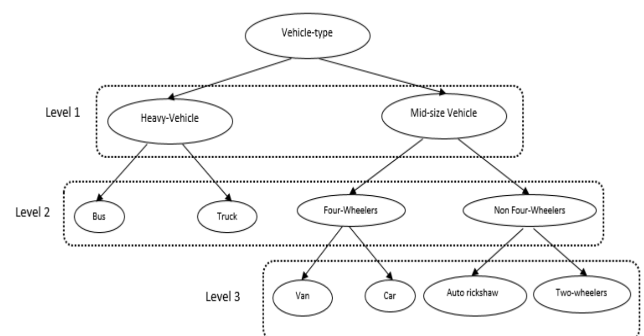


Fig. 3: Level 3 Hierarchy: Multi- Level Classifier

Here, in figure 2, a level 2 hierarchy, three classifiers are trained to train vehicle-type, heavy-vehicle and mid-size vehicle. The classifier of the parent node is trained using the train and validation images of that node's leaf node. That is, the vehicle-type node is trained using the sample images of the heavy-vehicle and mid-size vehicle nodes.

Figure 3 portraits level 3 hierarchy multi classifier. In level 3 hierarchy, all the classifiers are binary classifiers. We use a total of five classifiers in figure 3, to classify the vehicle to the correct class. For the binary classifier vehicle-type, all bus and truck inputs fitting to class heavy-vehicle are positive examples and all the input images fitting to class mid-size vehicle are negative. Similarly, for the binary classifier mid-size vehicle, input images van and car belonging to class four-wheelers are positive and input images auto rickshaw and two-wheelers belonging to class non-four wheelers are negative.
In single level classification, to classify an input image to class bus, the root node has to compare this input image with all the six leaf node classes. Therefore, in SLC, if there are n classes, every classification of input images takes n class comparison. Whereas, in MLC, at every levels, the number of classes get minimize. For example in fig, the binary classifier four-wheeler has only two classes to classify an instance belonging to leaf node van or car compared to six leaf nodes in single level classification. Therefore, for large set image classification problem, multi-level classification is best suitable as classification of an instance has to be differentiated between fewer classes and it also need fewer feature to train.

5. Dataset preparation

We use NITCAD dataset [24] to train single level and multi – level classifier to classify different vehicles in an unstructured road scene on an Indian road. For the implementation of our proposed model on NITCAD, the pedestrian class from the original dataset is not considered. The model is trained to classify the other six classes namely auto-rickshaw, bus, car, truck, two-wheelers and van. The images are cropped to extract individual classes and we use a subset of the labelled images from the dataset which consists of information about the location of various objects present in the corresponding frame. The images with the same content are avoided to reduce redundancy in the dataset. After this process, the subset of the dataset considered for implementation comprises 600 images with each class consisting of 100.
A limited dataset to train a deep learning model ends up in high risk of overfitting. So, we applied data augmentation such as gaussian blur, flipping, rotation, and gaussian noise to elaborate the data to fit in the dataset. Applying data augmentation introduced new images into our dataset by increasing the size of the dataset by a factor of four.

5.1 Evaluation metric

This section provides details of the evaluation metrics used to calculate the performance of the classification model.
We use classification accuracy, precision, recall and f1-score to evaluate the performance of the classifier. Classification Accuracy (A) is defined as the total number of images correctly classified, divided by total number of images within the dataset. It is mathematically defined as,

$$A = \frac{tp + tn}{tp + tn + fp + fn}$$

Where tp denotes true positives, tn denotes true negatives, fp denotes false positives, and fn denotes false negatives.
Precision P is the likeness of the percentage of correctly classified images to the total number of classified images. It is mathematically expressed as,

$$P = \frac{tp}{tp + fp}$$

tp denotes the correctly classified image and fp denotes the misclassified images.
The recall R is defined as the fraction of correctly classified images to the total number images. It is mathematically expresses as,

$$R = \frac{tp}{tp + fn}$$

F1-score is defined as the harmonic mean of precision and recall. It is mathematically expressed as,

$$f1 - score = 2 \times \frac{(P \times R)}{(P + R)}$$

6. Experimental Evaluation

We applied 80:20 ratio for training set and test set. For each classes the number of images trained are eighty and 20 are used for testing process. A limited dataset to train a deep learning model ends up in high risk of overfitting. So, we applied data augmentation such as gaussian blur, flipping, rotation, and gaussian noise to elaborate the data to fit in the dataset. Applying data augmentation introduced new images into our dataset by increasing the size of the dataset by a factor of four. Fig 2 shows the confusion matrix of single-level
classification. The overall accuracy for test image obtained id 85.21%
Similarly Fig 3 and 4 are confusion matrix obtained for 2 level hierarchy and 3 level hierarchy multi-level classifiers. The accuracy of 2 level hierarchy multi-level classifier is 87.92% and for 3 level hierarchy multi-level classifier is 92.92%

141

	Auto rickshaw	Bus	Car	Truck	Two−wheeler	Van
Auto rickshaw	72	2	4	3	4	5
Bus	0	65	0	6	2	0
Car	2	2	71	0	1	8
Truck	0	5	2	66	0	0
Two−wheeler	3	0	0	1	71	3
Van	3	6	3	4	2	64

Fig. 2: Confusion matrix of the single-level classification

	Auto rickshaw	Bus	Car	Truck	Two−wheeler	Van
Auto rickshaw	74	2	6	2	2	4
Bus	0	67	0	4	2	0
Car	1	2	70	0	1	5
Truck	0	4	2	69	0	0
Two−wheeler	3	0	1	2	74	3
Van	2	5	1	3	1	68

Fig. 3: Confusion matrix of the 2 level multi-level classification

	Auto rickshaw	Bus	Car	Truck	Two−wheeler	Van
Auto rickshaw	75	1	2	2	1	2
Bus	0	72	0	1	1	0
Car	1	2	75	0	1	2
Truck	0	2	0	73	0	0
Two−wheeler	3	0	2	2	77	2
Van	1	3	1	2	0	74

Table. 1: Performance evaluation of Single level classifier

Class	Accuracy	Error Rate	Precision	Recall	F1-Score
Auto - rickshaw	94.58	5.42	0.80	0.90	0.85
Bus	95.21	4.78	0.89	0.81	0.85
Car	95.42	4.58	0.85	0.89	0.87
Truck	95.63	4.37	0.90	0.82	0.86
Two -Wheeler	96.67	3.33	0.91	0.89	0.90
Van	92.92	7.08	0.78	0.80	0.79

Table. 2: Performance evaluation of 2 level hierarchy multi-level classifier

Class	Accuracy	Error Rate	Precision	Recall	F1-Score
Auto - rickshaw	95.42	4.58	0.82	0.93	0.87
Bus	96.04	3.96	0.92	0.84	0.88
Car	96.04	3.96	0.89	0.88	0.88
Truck	96.46	3.54	0.29	0.86	0.89
Two -Wheeler	96.88	3.12	0.89	0.93	0.91
Van	95	5	0.85	0.85	0.85

Table. 3: Performance evaluation of 3 level hierarchy multi-level classifier

Class	Accuracy	Error Rate	Precision	Recall	F1-Score
Auto - rickshaw	97.29	2.71	0.90	0.94	0.92
Bus	97.92	2.08	0.97	0.90	0.94
Car	97.71	2.29	0.93	0.94	0.93
Truck	98.13	1.87	0.97	0.91	0.94
Two -Wheeler	97.5	2.5	0.90	0.96	0.93
Van	97.29	2.71	0.91	0.93	0.92

7. Conclusion

In this paper, we propose a single and multi-level vehicle classification framework based on convolutional neural network and fine-tuned ResNet50. A new classification block is added to the ResNet50 through transfer learning to ensure generalization. The proposed model is trained on a subset of NITCAD object dataset with six categories of vehicle classes that are pertinent to unstructured environments. To alleviate overfitting and to increase the diversity of our dataset, we applied data augmentation techniques and observed the 3 level hierarch multi-level model perform better than 2 level hierarchy and single level classification.

References

[1]. M. Stocker, M. Rönkkö, and M. Kolehmainen, ''Situational knowledge representation for traffic observed by a pavement vibration sensor network,'' IEEE Trans. Intell. Transp. Syst., vol. 15, no. 4, pp. 1441–1450, Aug. 2014.

[2]. S.Meta and M.G.Cinsdikici, ''Vehicle-classification algorithm based on component analysis for single-loop inductive detector,'' IEEE Trans. Veh. Technol., vol. 59, no. 6, pp. 2795–2805, Jul. 2010.

[3]. S. A. Rajab, A. Mayeli, and H. H. Refai, ''Vehicle classification and accurate speed calculation using multi-element piezoelectric sensor,'' in Proc. IEEE Intell. Vehicles Symp. Proc., Jun. 2014, pp. 894–899.

[4]. P. K. Bhaskar and S. P. Yong, "Image processing-based vehicle detection and tracking method," in Proceedings of the 2014 International Conference on Computer and Information Sciences (ICCOINS), pp.1–5, IEEE, Kuala Lumpur, Malaysia, June 2014.

[5]. Y. U. Rong, W. A. N. G. Guoxiang, J. Zheng, W. A. N. G. Haiyan, "Urban road traffic condition pattern recognition based on support vector machine," Journal of Transportation Systems Engineering and Information Technology, vol. 13, no. 1, pp. 130–136, 2013.
[6]. L. T. Ng, S. A. Suandi, S. S. Teoh, "Vehicle classification using visual background extractor and multi-class support vector machines," in Proceedings of the 8th International Conference on Robotic, Vision, Signal Processing & Power Applications, pp. 221–227, Springer, Singapore, 2014.
[7]. Xinchen Liu, Wu Liu, Tao Mei, Huadong Ma, "PROVID: Progressive and Multimodal Vehicle Reidentification for Large-Scale Urban Surveillance", IEEE Transactions on Multimedia vol. 20, no. 3), pp. 645-658, 2018.

[8]. Z.Dong, Y.Wu, M.Pei, Y.Jia, "Vehicle type classification using a semi supervised convolutional neural network," IEEE Transactions on Intelligent Transportation Systems, vol. 16, no. 4, pp. 2247–2256, 2015.
[9]. Maria Crystal E. Orozco, Corazon B. Rebong, "Vehicular Detection and Classification for Intelligent Transportation System: A Deep Learning Approach Using Faster R-CNN Model", International journal of simulation: systems, science & technology, ISSN: 1473-804x, DOI 10.5013/IJSSST.a.20.S2.11.
[10]. X. Wang, W. Zhang, X. Wu, L. Xiao, Y. Qian, Z. Fang, "Real-time vehicle type classification with deep convolutional neural networks," Journal of Real-Time Image Processing, vol. 16, no. 1, pp. 5–14, 2019.
[11]. J. Cao, W. Wang, X. Wang, C. Li, J. Tang, "End-to-End view-aware vehicle classification via progressive CNN learning," in Proceedings of the CCF Chinese Conference on Computer Vision, pp. 729–737, Springer, Singapore, 2017.
[12]. C. Jiang and B. Zhang, "Weakly-supervised vehicle detection and classification by convolutional neural network," 2016 9th International Congress on Image and Signal Processing, Bio Medical Engineering and Informatics (CISP-BMEI), 2016, pp. 570-575, doi: 10.1109/CISP-BMEI.2016.7852775.
[13]. Krassimir Valev, Arne Schumann, Lars Sommer, and J ̈urgen Beyerer, "A Systematic Evaluation of Recent Deep Learning Architectures for Fine-Grained Vehicle Classification", arXiv: 1806.02987v1.
[14]. M. S. Chauhan, A. Singh, M. Khemka, A. Prateek, and R. Sen, "Embedded CNN based vehicle classification and counting in non-laned road traffic," in Proceedings of the Tenth International Conference on Information and Communication Technologies and Development, pp. 1–11, Ahmedabad, India, 2019.
[15]. S. Y. Jo, N. Ahn, Y. Lee, and S. J. Kang, "Transfer learning-based vehicle classification," in Proceedings of the 2018 International SoC Design Conference (ISOCC), pp. 127-128, IEEE, Daegu, South Korea, 2018.
[16]. Stephen Karungaru, Lyu Dongyang, and Kenji Terada, "Vehicle Detection and Type Classification Based on CNN-SVM," International Journal of Machine Learning and Computing, pp. 304 – 310Vol. 11, No. 4, July 2021.
[17]. Muhammad Atif Butt, Asad Masood Khattak, Sarmad Shafique, Bashir Hayat, Saima Abid, Ki-Il Kim, Muhammad Waqas Ayub, Ahthasham Sajid, and Awais Adnan, "Convolutional Neural Network Based Vehicle Classification in Adverse Illuminous Conditions for Intelligent Transportation Systems", Complexity, Hindawi, Volume 2021, https://doi.org/10.1155/2021/6644861.
[18]. Suraj Srinivas, Ravi K. Sarvadevabhatla, Konda R. Mopuri, Nikita Prabhu, Srinivas S.S. Kruthiventi, R. Venkatesh Babu, "An Introduction to Deep Convolutional Neural Nets for Computer Vision," Deep Learning for Medical Image Analysis, Academic Press, pp. 25-52, ISBN 9780128104088, 2017.

[19]. Połap D, "An adaptive genetic algorithm as a supporting mechanism for microscopy image analysis in a cascade of convolution neural networks." Applied Soft Computing 97:106824.
[20]. Z. Zhong, L. Zheng, G. Kang, S. Li, and Y. Yang, "Random erasing data augmentation," in Proceedings of the AAAI, pp. 13001–13008, 2020.

[21]. K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778, Seattle, WA, USA, 2016.
[22]. M.Xie, N.Jean, M.Burke, D.Lobell, and S.Ermon,"Transfer learning from deep features for remote sensing and poverty mapping," in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 30, Phoenix, AZ, USA, 2016.
[23]. J. Yue, W. Zhao, S. Mao, and H. Liu, "Spectral-spatial classification of hyperspectral images using deep convolutional neural networks," Remote Sensing Letters, vol. 6, no. 6, pp. 468–477, 2015.
[24]. Namburi GNVV Satya Sai Srinath, Athul Zac Joseph, S Umamaheswaran, Ch. Lakshmi Priyanka, Malavika Nair M, Praveen Sankaran, "NITCAD - Developing an object detection, classification and stereo vision dataset for autonomous navigation in Indian roads", in Proceedings of the Third International Conference on Computing and Network Communications (CoCoNet'19), pp.: 207–216, Procedia Computer Science 171, 2020.

[25]. M Swarnamugi and R Chinnaiyan, "Cloud and fog computing models for internet of things." International Journal for Research in Applied Science and Engineering Technology, December 2017.
[26]. Balachandar S., Chinnaiyan R. (2019) Reliable Digital Twin for Connected Footballer. In: Smys S., Bestak R., Chen JZ., Kotuliak I. (eds) International Conference on Computer Networks and Communication Technologies. Lecture Notes on Data Engineering and Communications Technologies, vol 15. Springer, Singapore
[27]. Balachandar S., Chinnaiyan R. (2019) Centralized Reliability and Security Management of Data in Internet of Things (IoT) with Rule Builder. In: Smys S., Bestak R., Chen JZ., Kotuliak I. (eds) International Conference on Computer Networks and Communication Technologies. Lecture Notes on Data Engineering and Communications Technologies, vol 15. Springer, Singapore
[28]. M. Swarnamugi and R. Chinnaiyan, "IoT Hybrid Computing Model for Intelligent Transportation System (ITS)," *2018 Second International Conference on Computing Methodologies and Communication (ICCMC)*, Erode, 2018, pp. 802-806

143

[29]. G. Sabarmathi and R. Chinnaiyan, "Reliable Machine Learning Approach to Predict Patient Satisfaction for Optimal Decision Making and Quality Health Care," *2019 International Conference on Communication and Electronics Systems (ICCES)*, Coimbatore, India, 2019, pp. 1489-1493
[30]. M. Swarnamugi and R. Chinnaiyan, "Smart and Reliable Transportation System based on Message Queuing Telemetry Transport Protocol," *2019 International Conference on Intelligent Computing and Control Systems (ICCS)*, Madurai, India, 2019, pp. 918-922
[31]. G. Sabarmathi and R. Chinnaiyan, "Investigations on big data features research challenges and applications," *2017 International Conference on Intelligent Computing and Control Systems (ICICCS)*, Madurai, 2017, pp. 782-786
[32]. G. Sabarmathi and R. Chinnaiyan, "Big Data Analytics Framework for Opinion Mining of Patient Health Care Experience," *2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC)*, Erode, India, 2020, pp. 352-357
[33]. Divya R., Chinnaiyan R. (2019) Reliable AI-Based Smart Sensors for Managing Irrigation Resources in Agriculture—A Review. In: Smys S., Bestak R., Chen JZ., Kotuliak I. (eds) International Conference on Computer Networks and Communication Technologies. Lecture Notes on Data Engineering and Communications Technologies, vol 15. Springer, Singapore.
[34]. Sabarmathi G., Chinnaiyan R. (2020) Envisagation and Analysis of Mosquito Borne Fevers: A Health Monitoring System by Envisagative Computing Using Big Data Analytics. In: Pandian A., Senjyu T., Islam S., Wang H. (eds) Proceeding of the International Conference on Computer Networks, Big Data and IoT (ICCBI - 2018). ICCBI 2018. Lecture Notes on Data Engineering and Communications Technologies, vol 31. Springer, Cham.
[35]. Swarnamugi M., Chinnaiyan R. (2020) Context—Aware Smart Reliable Service Model for Intelligent Transportation System Based on Ontology. In: Singh P., Kar A., Singh Y., Kolekar M., Tanwar S. (eds) Proceedings of ICRIC 2019. Lecture Notes in Electrical Engineering, vol 597. Springer, Cham
[36]. R.Chinnaiyan and S.Balachandar BDET 2020: Proceedings of the 2020 2nd International Conference on Big Data Engineering and Technology, January 2020 Pages 106–111
[37]. R.Chinnaiyan, S.Somasundaram(2010) "Evaluating the Reliability of Component Based Software Systems " ,International Journal of Quality and Reliability Management , Vol. 27, No. 1., pp. 78-88
[38]. R.Chinnaiyan, S.Somasundaram(2011), "An Experimental Study on Reliability Estimation of GNU Compiler Components - A Review", International Journal of Computer Applications,Vol.25, No.3, July 2011, pp.13-16.
[39]. R.Chinnaiyan, S.Somasundaram(2009) "Reliability of Object Oriented Software Systems using communication variables – a review " ,International Journal of Software Engineering, Vol.2, No.2,PP.87-

AUTHORS PROFILE

M. Swarnamugi, completed her master's in computer science and engineering from Pondicherry University. Presently pursuing P.hD. computer applications from Visvesvaraya Tecnological University, Belgavi. Her research interest include artificial intelligence, IoT and Intelligent systems.

Dr. R. Chinnaiyan, completed his Ph.D in Computer Science in Anna University-Chennai in 2012. Currently he is guiding 4 research scholars in VTU. He is a life member of Indian Society of Technical Education, ACM and Computer Society of India. He had published 65+ research papers in reputed National and International Conferences and Journals. His research interest include Internet of Things, Big Data Analytics, Cloud Security and Reliability of Wireless Sensor Networks.