Journal of Ambient Intelligence and Humanized Computing
https://doi.org/10.1007/s12652-021-02924-4

ORIGINAL RESEARCH

Hadoop Deep Neural Network for offending drivers

Shiva Asadianfam[1] · Mahboubeh Shamsi[2] · Abdolreza Rasouli Kenari[2]

Received: 26 March 2020 / Accepted: 20 January 2021

Abstract
Deep learning is recently regarded as the closest artificial intelligence model to human brain. It is about learning multiple levels of representation and abstraction that help to make sense of data such as images, sound, and text. Based on MapReduce framework and Hadoop distributed file system, this paper proposes a distributed approach for detect offending drivers and training the Deep Neural Network models such as Convolutional Neural Network (CNN) and Long Short Term Memory network (LSTM). Its implementation and performance are evaluated on Big Data platform Hadoop. The intelligence growing process of human brain requires learning from Big Data. The main contribution of this paper is that it is implemented to analyze traffic big data and to detect offending drivers in Hadoop by CNN with Support Vector Machine (SVM) and LSTM. The efficiency of the proposed method is computed by using experimental and theoretical analysis.

Keywords Deep learning · Object detection · Deep Neural Network · Convolutional Neural Network · Hadoop · Long Short Term Memory network

1 Introduction

1.1 Research motivation and challenges

The development of cities and the changing pattern of people's movement patterns have accelerated indescribably as human life becomes more complex to meet their daily needs, to the point that in all countries of the world, the personal car as the main element of the family plays a key role in development (Moghaddam and Ayati 2014). Similarly, the physical and mental consequences of the presence of this new element are so widespread and harmful that in some countries of the world, the number of people killed as a result of traffic accidents is more than those killed due to deadly diseases (Rahemi et al. 2017; Tao et al. 2017). Vehicles are a feature of civilization, and driving is essential for various activities of contemporary life, such as work, social life, recreation, socio-educational activities, and the like. The need for widespread use of vehicles in everyday life has led to an increase in the number of road and urban accidents with frequent and high mortality rates, which has become a major problem in various social areas (Lotfi 2012; Rakotonirainy et al. 2014).

Today, issues and problems in transportation and urban traffic have become a serious problem. The importance of these cases justifies the use of urban traffic monitoring systems for today's societies. Various technologies are used in urban traffic monitoring systems, and recently, despite the volume, variety and speed of data production in these systems (Chen et al. 2014a, b), the use of big data technology is increasing rapidly. Automatic analysis of traffic situation and its intelligent and optimal control based on the analysis is a new and important solution in these systems (Aoyama 1997; McLauchlan et al. 1997; Park et al. 1999). Big data is a term used to describe a large body of information that also grows exponentially over time. In short, such data is so large and complex that none of the conventional data management tools can store or process it effectively (Chen et al. 2014a, b). Hadoop is a popular technology for big data processing. Hadoop and the MapReduce techniques are utilized in various fields, such as behavior detection. MapReduce is a programming model for large-scale data processing, which

✉ Mahboubeh Shamsi
shamsi@qut.ac.ir

Shiva Asadianfam
sh_asadianfam@yahoo.com

Abdolreza Rasouli Kenari
rasouli@qut.ac.ir

1 Department of Computer Engineering, Qom Branch, Islamic Azad University, Qom, Iran

2 Faculty of Electrical and Computer Engineering, Qom University of Technology, Qom, Iran

Published online: 04 February 2021

reduces processing time using distributed data processing. Therefore, this paper presents a Hadoop Deep Neural Network for offending drivers.

1.2 Our approach

The Hadoop Deep Neural Network structure is proposed for image labeled and classification in this study. To our knowledge; this is the first attempt to use a CNN–LSTM structure in traffic image processing. Under the scope of the study, images of 10,000 offending drivers that were obtained from traffic control center were. To improve classification of CNN, we used Support Vector Machine (SVM) classifiers along with CNN. The experimental results were compared. The hybrid structure in this study is proposed as an image classification method with high classification performance.

The main challenge in this study is to identify the offending drivers'. This paper extracts some factors such as objects (vehicles), location, resources (including traffic signs), nearby object and nearby action by segmentation of the input images. Then, by using two types of deep learning algorithms along with the MapReduce technique on five input images can be identified the offending drivers.

1.3 Organization of the paper

The rest of paper is organized as follows: In Sect. 2, an overview of big data and deep learning is presented. In Sect. 3, some related articles based on traffic big data and deep learning methods are reviewed. In Sect. 4, the proposed system and the process of deep learning are explained. In Sect. 5, the experimental results are discussed, and finally, in Sect. 6, conclusion of the overall research work is given.

2 An overview of Big Data and deep learning

2.1 Big Data

Big Data is a collection of data that is huge in volume, yet growing exponentially with time. It is a data with so large size and complexity that none of traditional data management tools can store it or process it efficiently. Big data is also a data but with huge size (Secundo et al. 2017; De Mauro et al. 2018). Big data may be obtained from social media, traffic and transportation, industry, medical care, public organizations, families, and so on (Gantz and Reinsel 2011). Big data can be described by (a) volume, (b) variety, (c) velocity,(d)variability characteristics(Laney 2001). The big data value chain can be divided into four stages: data production, data acquisition, data storage, and data analysis (Chen et al. 2014a, b). Big data production refers to the main source of big data, which includes the data collected in a variety of fields via different technologies. The big data acquisition stage involves data collection, data transfer to storage infrastructure, and data preprocessing. In the data storage stage, various mechanisms (such as storage on the HDFS) and certain types of programming model (like MapReduce) are employed. In the big data analysis stage, the methods related to big data analysis are employed (Chen et al. 2014a, b). MapReduce is a software framework and programming model used for processing huge amounts of data. MapReduce program works in two phases, namely, Map and Reduce. Map tasks deal with splitting and mapping of data while Reduce tasks shuffle and reduce the data. Hadoop is capable of running MapReduce programs written in various languages. Hadoop allows us to run applications on thousands of nodes with thousands of terabytes of data (White 2012). Hadoop consists of two major sections, namely MapReduce and HDFS. This system is actually designed to run on multiple servers. The MapReduce section and the HDFS section run on the main server and the secondary servers, respectively. The Hadoop Distribution File System (HDFS) (Shvachko et al. 2010) is able to increase the rate of data transfer between nodes and enables the system to continue operating incessantly in case of the failure of one node. This approach actually reduces the risk of catastrophic system failure even when a significant number of nodes are deactivated(Dean and Ghemawat 2008). Figure 1 illustrates an overall diagram of the big data and MapReduce technique. An input (big data) to a MapReduce in Big Data job is divided into fixed-size pieces called input splits. Input split is a chunk of the input that is consumed by a single map. In Map phase data in each split is passed to a mapping function to produce output values. Shuffling phase consumes the output of Map phase. Its task is to consolidate the relevant records from Map phase output. In Reduce phase, output values from the Shuffling phase are aggregated. This phase combines values from Shuffling phase and returns a single output value (White 2012).

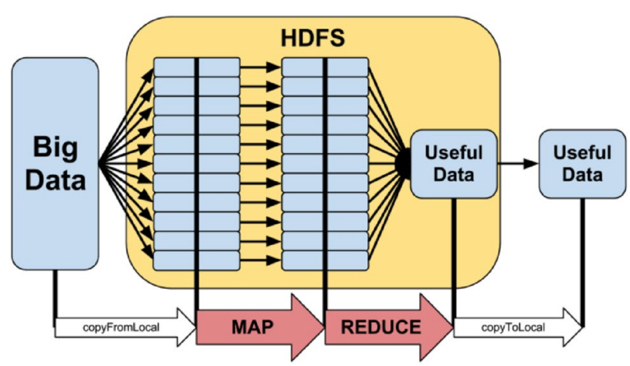


Fig. 1 Big data and MapReduce technique

2.2 Deep learning

Deep learning is an artificial intelligence function that imitates the workings of the human brain in processing data and creating patterns for use in decision making. Deep learning is a subset of machine learning in artificial intelligence that has networks capable of learning unsupervised from data that is unstructured or unlabeled. Also known as deep neural learning or Deep Neural Network (Goodfellow et al. 2016). Deep learning has evolved hand-in-hand with the digital era, which has brought about an explosion of data in all forms and from every region of the world. This data, known simply as big data, is drawn from sources like social media, internet search engines, e-commerce platforms, and online cinemas, among others. This enormous amount of data is readily accessible and can be shared through fintech applications like cloud computing. However, the data, which normally is unstructured, is so vast that it could take decades for humans to comprehend it and extract relevant information. Companies realize the incredible potential that can result from unraveling this wealth of information and are increasingly adapting to AI systems for automated support (Schmidhuber 2015).

In general, deep learning methods are divided into four different categories based on the base method from which they are derived: Convolutional Neural Networks, Restricted Boltzmann Machines (RBMS), Autoencoders, and Sparse Coding. For each of these categories, certain architectures are proposed. For example, in CNNs, there are Alexnet, SPP, VGGnet, Clarfai and Googlenet architectures which are often different in terms of the number of their convolutional and fully-connected layers. Learning methods vary based on their features, such as Generalization, Unsupervised Learning, Feature Learning, Real-Time Training, Real-Time Prediction, Biological understanding, Theoretical justification, Invariance, and Small training set (Patterson and Gibson 2017).

2.2.1 Convolutional Neural Networks

A Convolutional Neural Network (CNN) is a Deep Learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other. CNNs were inspired by earlier work that showed that the visual cortex in animals contains complex arrangements of cells, responsible for detecting light in small local regions of the visual field (Hubel and Wiesel 1968). CNNs were developed in the 1980s and have been applied to image, speech, text, and drug discovery problems (Atlas et al. 1988; LeCun et al. 1989, 2015). A predecessor to CNNs was the Neocognitron (Fukushima and Miyake 1982). A typical CNN is composed of convolutional and dense layers (Fig. 2). The purpose of the first convolutional layer is the extraction of common patterns found within local regions of the input images. CNNs convolve learned filters over the input image, computing the inner product at every image location in the image and outputting the result as tensors whose depth is the number of filters.

The pre-processing required in a CNN is much lower as compared to other classification algorithms. While in primitive methods filters are hand-engineered, with enough training, CNN have the ability to learn these filters/characteristics. The architecture of a CNN is analogous to that of the connectivity pattern of Neurons in the Human Brain and was inspired by the organization of the Visual Cortex. Individual neurons respond to stimuli only in a restricted region of the visual field known as the Receptive Field. A collection of such fields overlap to cover the entire visual area (Wallach et al. 2015).

2.2.2 Long Short Term Memory networks (LSTM)

Long Short Term Memory networks (LSTM) are a special kind of Recurrent Neural Network (RNN), capable of

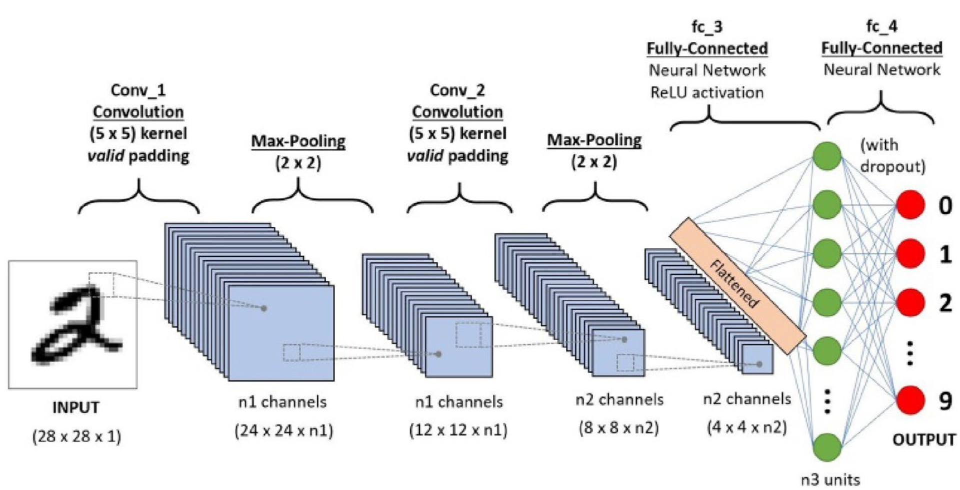


Fig. 2 Convolutional Neural Network model

learning long-term dependencies. They were introduced by (Hochreiter and Schmidhuber 1997), and were refined and popularized by many people in following work. They work tremendously well on a large variety of problems, and are now widely used. LSTMs are explicitly designed to avoid the long-term dependency problem. Remembering information for long periods of time is practically their default behavior, not something they struggle to learn. All RNNs have the form of a chain of repeating modules of neural network. In standard RNNs, this repeating module will have a very simple structure, such as a single tanh layer. LSTMs also have this chain like structure, but the repeating module has a different structure. Instead of having a single neural network layer, there are four, interacting in a very special way.

On the other hand, LSTM networks are a modified version of recurrent neural networks, which makes it easier to remember past data in memory. The vanishing gradient problem of RNN is resolved here. LSTM is well-suited to classify images (Dvornek et al. 2017; Kutlu and Avcı 2019), process and predict time series given time lags of unknown duration. It trains the model by using back-propagation. In Fig. 3, each line carries an entire vector, from the output of one node to the inputs of others. The pink circles represent pointwise operations, like vector addition, while the yellow boxes are learned neural network layers. Lines merging denote concatenation, while a line forking denote its content being copied and the copies going to different locations.

3 Related works

A new data analysis framework is available for smart cities (Osman 2018), discussing the approaches to knowledge discovery in traditional systems versus those in big data systems. Student behavior detection based on the big data framework is explored (Cantabella et al. 2019). The Monte Carlo simulation-based forecast of traffic speed was done using big data history (Jeon and Hong 2016). In article (Yin et al. 2015), big data issued for pedestrian counting. Yin simulated new data sources, such as 'big data', and computational analysis on pedestrians (Yin et al. 2015).

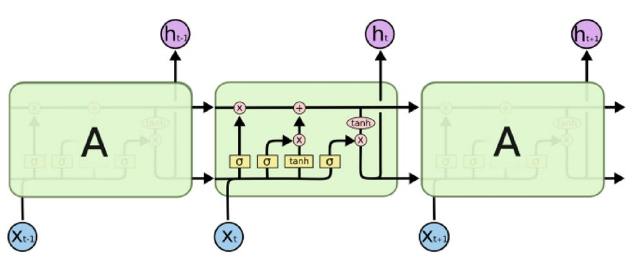


Fig. 3 Long Short Term Memory networks model

Asadianfam and et al. introduce a big data system for analyzing unsafe behaviors based on images and texts (Asadianfam et al. 2020a, b). Lu presents a big data platform for identifying the unauthorized behaviors of urban drivers in construction waste discharge in Hong Kong in order to manage construction waste and combat urban crimes (Lu 2019). Guo et al. introduce a big data-based platform for identifying workers' safe behaviors in a metro construction project in China (Guo et al. 2016).

In the area of traffic control, the following studies have been performed using deep learning. In paper (Yu et al. 2016), a new algorithm based on deep models is proposed to detect traffic signs. The results of implementing this algorithm in paper (Yu et al. 2016) indicate over 97% accuracy in detection. In Paper (Arcos-García et al. 2018a, b) investigates several systems for detecting traffic signs via Deep Neural Networks and analyzes the memory allocation, accuracy, and processing time of each system. In paper (Arcos-García et al. 2017), deep learning is used for the traffic sign detection systems with approximately 100 percent precision. Also, in paper (Arcos-García et al. 2018a, b), deep learning is employed to spot traffic signs. The automatic detection of traffic lights through deep learning classification techniques based on GPS tracking during driving is discussed in paper (Munoz-Organero et al. 2018). In paper (Rao et al. 2019) a distracted driving recognition method based on deep Convolutional Neural Network is proposed for the driving image data. A multi-layer CNN network is constructed in the model and the key parameters of the input layer, convolution layer, pooling layer, fully connected layer and output layer are optimized as well. In paper (Kim 2019) proposes a traffic monitoring system that detects, tracks, and classifies multiple vehicles on the road. Kim is used the process of machine learning based on a Convolutional Neural Network (CNN). In paper (Kumar and Anuradha 2020) introduced the method namely Enhanced Convolution neural network with Support Vector Machine based vehicle detection. The efficiency of the proposed method is computed by using experimental and theoretical analysis. In paper (Asadianfam et al. 2020a, b), CNN is used to offending drivers detection. The distributed deep learning system was implemented to analyze traffic big data and to detect driver violations. To our knowledge, in the realm of Traffic control, deep learning has been mostly employed in the detection of traffic signs. The present study attempts to examine Convolutional Neural Network and Long Short Term Memory networks along with the MapReduce technique over big traffic to identify offending drivers. A review of the relevant studies on traffic big data using MapReduce technique and deep learning methods with their advantages, disadvantages are provided in Table 1.

Table 1 Some related articles based on traffic big data and deep learning methods

Area	References	Description	Advantage	Disadvantage
Traffic big data	Osman (2018)	Is presented a MapReduce-based K-means algorithm	The introduction of a big data high-speed framework required a little time	Not analyzing performance metrics
	Cantabella et al. (2019)	Apriori algorithm based on MapReduce	Processing student data at a high speed	Not computing precision of analysis
	Jeon and Hong (2016)	The Monte Carlo simulation using big data	Traffic speed predictions using big data	–
	Yin et al. (2015)	Big Image data for recognition ACF algorithm	Analysis of the number of pedestrians at a high speed	Not analyzing performance metrics
	Asadianfam et al. (2020a, b)	The real-time analysis of images and texts	Processing drivers data at a high speed	Not enough accuracy
	Lu (2019)	Big data analytics to identify illegal construction waste dumping	Analysis of illegal construction	Not analyzing processing time reduction
	Guo et al. (2016)	A Big-Data-based platform of workers' behavior	Analysis of workers' behavior	Not analyzing performance metrics
Deep learning	Yu et al. (2016)	Detection of traffic signs	Over 97% accuracy	–
	Arcos-García et al. (2018a, b)	Detection of traffic signs	Analyzes the memory allocation, accuracy, and processing time of each system	Deep learning is employed to spot traffic signs
	Munoz-Organero et al. (2018)	Detection of traffic lights	Analyzes the allocation, accuracy	GPS tracking during driving
	Rao et al. (2019)	A distracted driving recognition method	Multi-layer CNN network	–
	Kim (2019)	Detect multiple vehicles on the road	Based on CNN	Not analyzing processing time
	Kumar and Anuradha (2020)	Vehicle detection	Introduced the method namely Enhanced Convolution neural network with Support Vector Machine	Not analyzing processing time
	Asadianfam et al. (2020a, b)	traffic violation detection	Analyzes the accuracy, and processing time and other factors	Only CNN

4 The proposed approach

The present study mainly focuses on using MapReduce technique to detect offending drivers on deep learning approach. The proposed system is designed in such a way that it would be able to detect risky behaviors of offending drivers in big data using the distribution technique and deep learning algorithms. The proposed system with a MapReduce space uses deep learning for a wide variety of data. This system is able to analyze the data obtained from traffic control centers in a distributed manner and it is capable to detect of offending drivers. As you can see in Fig. 4, five factors (Goh and Chua 2009), such as objects (vehicles), location, resources (including traffic signs), nearby object and nearby action are used to extract the drivers' behavior based on the input images. Then, by using two types of deep learning algorithms along with the MapReduce technique on five input images (dividing the input image into five segments) can be identified the offending drivers.

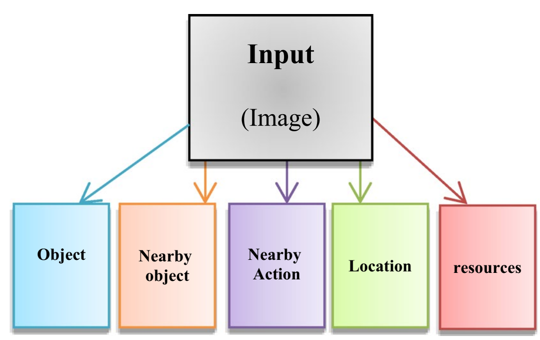


Fig. 4 The segmentation of input big data

The overall process of MapReduce offending drivers' detection with CNN is depicted in detail in Fig. 5. This figure illustrates the overall architecture of the deep Convolutional Neural Networks for driver violation detection in different layers. First, the main image is divided into

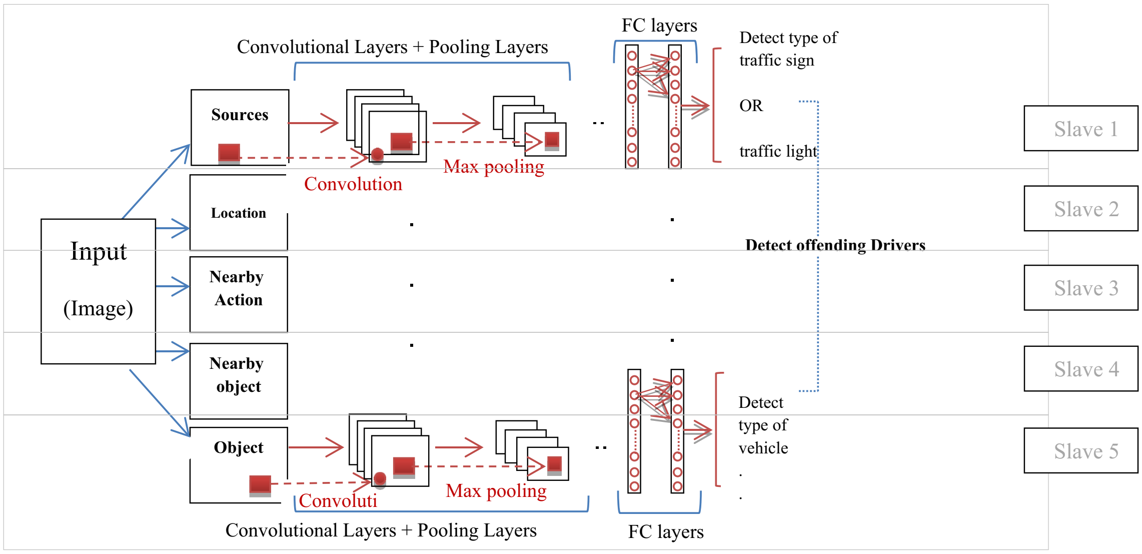


Fig. 5 The overall MapReduce processes of proposed system by CNN

several images including the predefined factors of driving violation (the resources, location, nearby action, nearby object, and vehicle), and then based on the layered training data, each of the deep nervous networks is trained. Finally, each of them generates a multilayer network with convolutional and connected layers. In this study, the Convolutional Neural Networks (Aghdam and Heravi 2017) are of VGGnet type consisting of 13 to 15 convolutional layers and three fully-connected layers. Its convolutional layers and pooling layers are located alternately, and after these layers, there are three fully-connected layers. This provides a thorough incrementally deep evaluation of the network. Each of the deep Convolutional Neural Networks learns the features in the driver violation images, such as traffic signs, traffic lights, street areas, vehicles, etc., and extracts the required features for categorizing the images. The combination of the output of each Convolutional Neural Network is considered as offending drivers. On the other hand, initially, the images obtained from the traffic control centers are divided on the slave nodes. Each slave node analyzes the image independently. Therefore, in the Map phase, in each slave node, the functions of traffic sign detection, traffic lights, street areas, vehicles, etc. are run using deep learning. In the Reduce phase, different outputs of each of these systems for the input images are combined and a label, as the violation found in the image, is introduced. There are a number of deep learning-related methods; however, in this paper, the Convolutional Neural Networks are used. In general terms, a CNN consists of three main layers: the convolutional layer, the pooling layer, and the fully-connected layer. In each neural network, there are two stages, namely, the Feed-Forward stage and the Back-Propagation stage. The Back-Propagation stage is for training and Feed-Forward stage is for prediction. The input image is fed into the network, which is done via the dot product between the input and the parameters of each neuron, and finally, the convolutional operations are done in each layer. In the proposed structure, a convolutional network is shared among all the pixels of the input image. Its weight sharing significantly reduces the number of the trainable free parameters of the network, and as a result, increases generalizability. Then, the network output is calculated.

The process of MapReduce offending drivers' detection with LSTM is described in Fig. 6. A long Short Term Memory network is used to describe the current state of the image by analyzing the current state of the image. Each slave run separate LSTM network. Input of each LSTM network is one of segmentation in Fig. 4. End of each phase is used CNN along with LSTM network. The key to LSTMs is the cell state, the horizontal line running through the top of the diagram. The cell state is kind of like a conveyor belt. It runs straight down the entire chain, with only some minor linear interactions. It's very easy for information to just flow along it unchanged. The LSTM does have the ability to remove or add information to the cell state, carefully regulated by structures called gates. Gates are a way to optionally let information through. They are composed out of a sigmoid neural net layer and a pointwise multiplication operation. The sigmoid layer outputs numbers between zero and one, describing how much of each component should be let through. A value of zero means "let nothing through," while a value of one means "let everything through!". An LSTM has three of these gates, to protect and control the cell state.

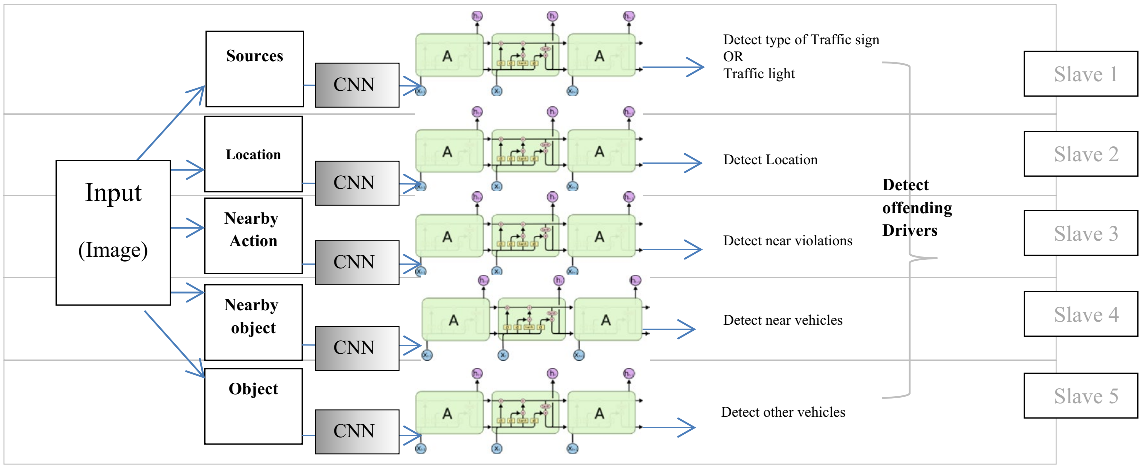


Fig. 6 The overall MapReduce processes of proposed system by using LSTM

5 Results and discussion

In this section, the efficiency of the proposed approach is examined and the results are presented. In this experimental study, the proposed approach is implemented via Java language along with OpenCV library. The tested computer system is implemented in the VMware environment using two clusters. The first cluster has only one node and the second cluster contain a master node and seven slave nodes. The Linux Ubuntu 14.04 LTS operating system, Hadoop 2.7.1 is installed and run on all the nodes. In order to investigate and test the proposed system, the information provided by the cameras in the city is utilized. In the present study, images captured by the traffic control center were 100,000 records and predefined behavioral data of traffic were 171 records including action, object, location, sources, nearby object and nearby action. Various criteria in analyzing the results of proposed system, such as performance and scalability, cost, efficiency and accuracy are described in detail the following sections.

5.1 Performance and scalability

To compare the performance of proposed system in the sequential program without the MapReduce technique and in Hadoop in stand-alone mode, CPU time of a number of images were examined in the range of 1000–10,000. CPU time was calculated from the time images were received to the time the output of proposed system was obtained (Table 2). The analysis of CPU time for images data, CPU time increases as the number of multimedia data (the images) augments. For example, in the processing of 100 images via Hadoop in stand-alone mode, CPU time is 3300 units while in the processing of 10,000 images, it increases to 33,000 units.

Based on the results presented in Fig. 7, apart from the single node cluster (Hadoop in stand-alone mode), the system has near-linear scalability. However, in the analysis of images data, compared to that of the Hadoop cluster with the one slave node, the Hadoop cluster with seven slave nodes leads to an almost 70% reduction in CPU time. The reason for this improvement is that in the MapReduce program, the system automatically performs calculations among large clusters of machines in parallel, manages system failures, and schedules within-system communication, and in this way, it makes the use of network and disks easier and more effective. Thus, through the balanced distribution of computing load across nodes, the parallel execution is performed without interference and CPU time decreases.

5.2 Cost of Hadoop clusters

This scenario is designed to estimate how much cost of resources might be achieved in sequential program and proposed system during the analysis with virtual

Table 2 CPU time of camera images in the sequential program and Hadoop in stand-alone mode

Mode	Number of images									
	1000	2000	3000	4000	5000	6000	7000	8000	9000	10,000
Sequential program	3000	6000	9300	12,200	15,000	17,970	20,850	24,800	27,500	30,000
Hadoop stand-alone	3300	6400	9300	12,400	15,800	18,100	21,000	25,000	28,400	33,000

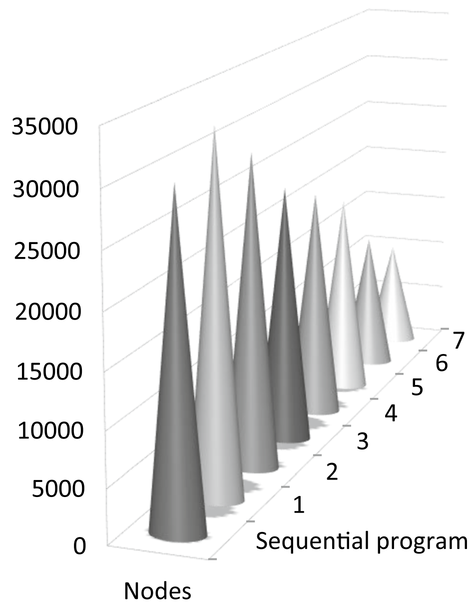


Fig. 7 CPU time using proposed system with regard to the number of nodes

machines has been used to build up Hadoop clusters. It was also important in this analysis to compare the same amount and the same strong virtual machines within the Hadoop cluster. We used Hadoop version 2.7.1 on Linux Ubuntu 14.04 LTS, with the weakest category of VMware Workstation. This cluster has number of slave nodes (from 1 to 7), each with 7 core, 2 GB of RAM, 3 hard drives. This cluster has an estimated cost of about 800$ (Memory) and 1200$ (Hardware). Figure 8 shows the estimated costs of the Hadoop cluster consisting of slave nodes (from 1 to 7) and sequential program.

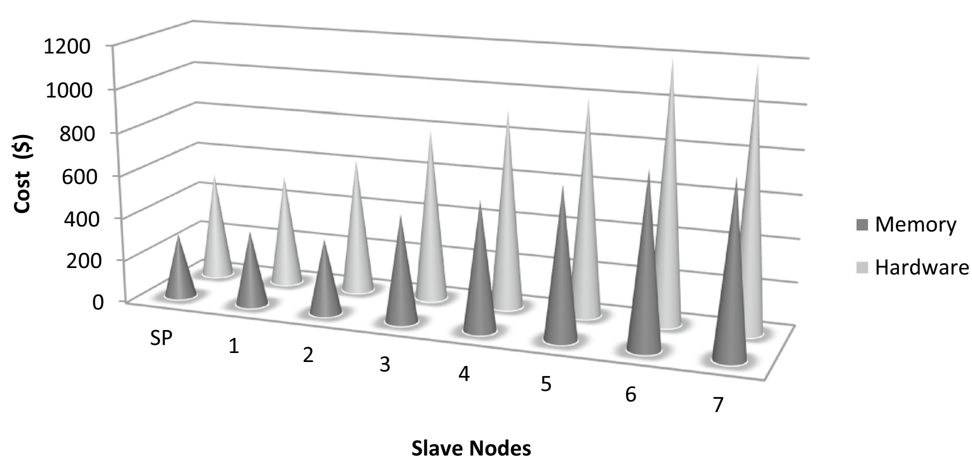
5.3 Efficiency in the Hadoop cluster using deep learning

5.3.1 Convolutional Neural Network

In order to investigate the proposed system with a deep learning approach, a sample of the data gathered from the surveillance cameras is depicted in Fig. 9. As shown in Fig. 9, the input data obtained from the traffic surveillance cameras are first divided into several segments. In the analysis of surveillance camera images, the image fragmentation is done based on the effective factors which are derived from expert knowledge. The image fragmentation is used to find the location of the traffic signs, traffic lights, vehicle, nearby object, and nearby action. In the Map phase, a Convolutional Neural Network is selected for each segmented image based on the type of function that will be used in the next phase. For instance, the first convolutional network would be able to identify the type of category pertaining to traffic signs or traffic lights. As you can see in Fig. 8, in each layer, a Convolutional Neural Network (CNN) arranges its neurons in three dimensions (width, height, and depth). Each layer converts an input CNN network into a three-dimensional mass into a three-dimensional mass. Next converts the output of the neuron activation values. In this example, each of the five fragmented images contains pixel values of the image, so its width and height will be the dimensions of the image and its depth will be equal to 3 (red, green and blue channels related to the image). In the Reduce phase, via the combination of the results of the convolutional networks, the driver violation is detected and displayed.

Because of the high-dimension images, it is impractical to connect the neurons to all of the neurons behind them. Therefore, each neuron is connected only to a small area of the input image. The dimensions of this small area are big enough for a hyper parameter to be connected to them. This area is called Receptive Field. Table 3 shows the training

Fig. 8 Estimated costs of the Hadoop cluster



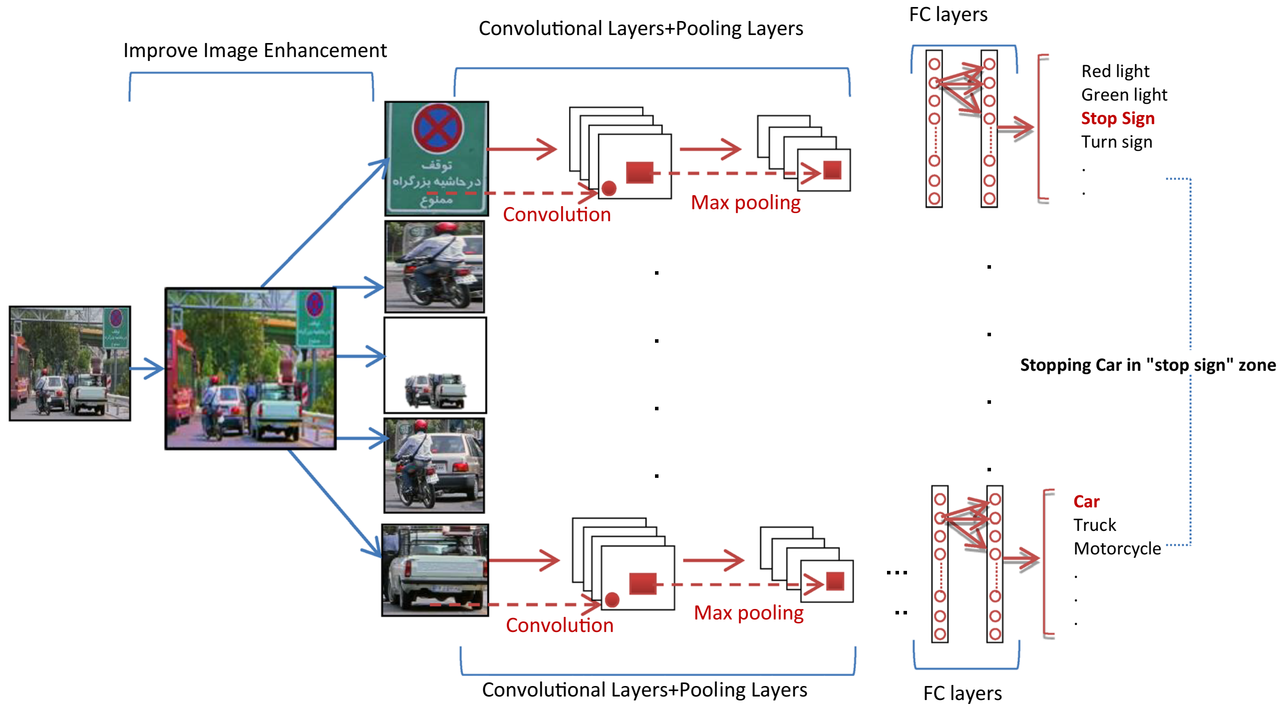


Fig. 9 A sample test data using the distributed convolutional neural network

Table 3 The experimental results for different receptive fields

No	Receptive field	Weight	Training time (s)
			341
2	4×4	48	547
3	5×5	75	710
4	6×6	108	878
5	7×7	147	906

time for different receptive fields with different dimensions. For example, the second row in Table 3, implies that each neuron in the convolutional layer has a weighting factor of 48 for a 4×4×3 (length, width, and depth) receptive field in the input image. Therefore, the dot product of the weights and input images is performed, the results are passed through a nonlinear function, the connections are locally made with regard to the location, and the output is generated. For the training data obtained from the images in the unstructured police data with two slave nodes, the learning time of the 4×4 receptive field is 1189 units. As can be observed, the learning time increases as the area of the receptive field (of the model) increases. Therefore, it can be declared that the distributed violation detection algorithm requires more learning time, and that this time increases linearly with an increase in the area of the receptive field, which makes it suitable for large-scale learning.

Activations in an example of the deep learning approach are shown in Fig. 10. The initial mass stores the raw pixels of the image (car image on the left) and the last mass holds the category points (fully connected layer). Each activation mass is displayed as a column during the processing path. We arranged the slices of each mass in rows, because it is difficult to display 3D masses. The mass of the last layer contains the scores of each class. But here we showed only 5 points higher. Suppose a neural network categorizes video convolution as a car image. How can we be sure that the network is actually choosing the car and not the content information from the background or other objects in the image? In other words, how do we make sure that the network really identifies the car and that the answer to its categorization is not simply because of other information in the image? One way to detect network prediction is to plot the probability of the desired category (here the category of cars) as a function of the location of a blocked object. This means that we move on different areas of the image and put a patch on the image that makes the whole area equal to zero, and then look at the probability of the class. We can display the probability as a two-dimensional heat map.

5.3.2 Long Short Term Memory networks along with CNN

LSTM is well-suited to classify images, process and predict caption of images. Rapid classification of images that are detected in the traffic images is of great importance in

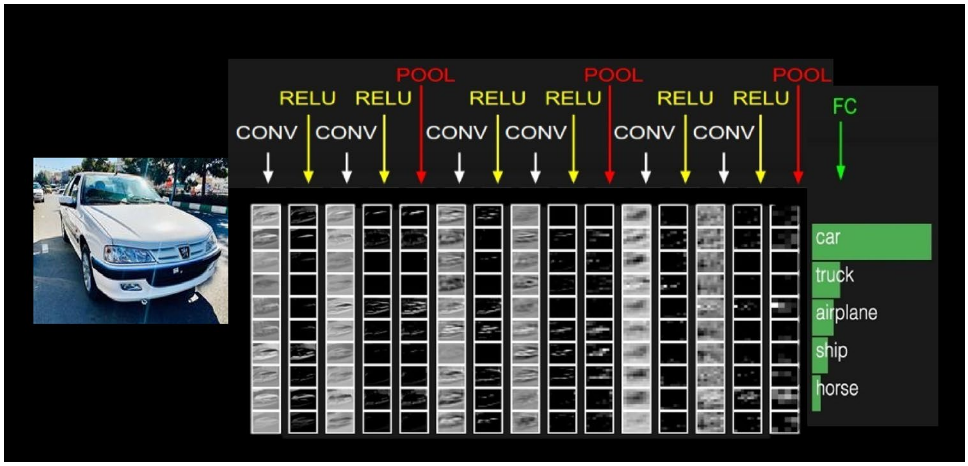


Fig. 10 Activations in an example of the distributed convolutional neural network

the early offending drivers. Moreover, in order to compare the performance of the proposed system in different conditions, CPU time of the surveillance camera images using the MapReduce programming model is shown in Fig. 5, the deep distributed Convolutional Neural Network is proposed in Fig. 6, and a different number of slaves are also explored. The results achieved from deep learning and the detection functions in the MapReduce programming model displayed in Figs. 5 and 6 are given in Table 4.

In Table 4, Map Time and Reduce Time specify CPU times of the Map function and the Reduce function, respectively. Obviously, without the MapReduce technique, no time would be available to process the Map and Reduce functions. CPU Time is the total Job processing time consisting of loading, shuffling, and etc. Compared to the sequential program, Hadoop with a single node required more time for large-scale data processing because of its high overhead. In addition, CPU time of the Map function decreases dramatically by increasing the number of nodes using deep learning. For example, in the processing of images, CPU time of the program using CNN for the training data of the Hadoop cluster with two nodes is equal to 31,099 units; however, CPU time reduces to 30,147 units when LSTM network is employed. By increasing the number of nodes to 4 using CNN on each node, CPU time decreases to 23,080. Based on the results reported in Table 4, it can be concluded that the CPU time for each node decreases as much as the number of slave' rises due to its parallel execution, and over the course of training through deep learning, since the training time is also taken into account, the total time increases. The Map time is usually high owing to the distribution of the data on the slaves, and the CPU time is reported considering the MapReduce time. In addition, the MapReduce technique is more efficient when the number of samples increases. So, it is required to strike a balance between the number of slaves and the volume of the data set in order to obtain more efficient results.

Table 5 shows the training time and accuracy for different dimensions. From the experimental results, the accuracy decreases when the dimension increases. The reason is that the image with more pixels requires more hidden features to represent their characteristics. If the number of hidden nodes is not enough, the abstracted patterns are inadequate to represent those images, so that the accuracy is very low. For example, the accuracy drops to around 9.64% for 10×10 dimensions when only 100

Table 4 The MapReduce and processing times using the methods presented in Fig. 5 and Fig. 6

	Time	Sequential program	# Nodes (seconds)			
			1	2	3	4
Deep learning algorithm 1-CNN (Fig. 5)	Map (s)	–	27,201	24,320	18,222	17,118
	Reduce (s)	–	6809	6679	6783	5962
	CPU (s)	33,810	34,010	**31,099**	25,005	**23,080**
Deep learning algorithm 2–LSTM (Fig. 6)	Map (s)	–	24,521	22,666	17,463	15,347
	Reduce (s)	–	6412	5972	6900	4123
	CPU (s)	33,810	33,780	**30,147**	23,000	19,142

Table 5 Experiment results for different dimensions

Dimension	Number of cases	Number of hidden units	Training time (ms)	Accuracy
3×3	17	100	6265	99.98% ± 0.01%
4×4	31	100	8123	99.98% ± 0.01%
5×5	57	100	10,541	99.97% ± 0.03%
6×6	131	100	12,947	99.25% ± 0.75%
7×7	216	100	15,891	96.38% ± 4.42%
8×8	498	100	20,563	72.00% ± 6.08%
9×9	1003	100	23,547	35.27% ± 4.21%
10×10	1940	100	25,497	9.64% ± 3.01%

Table 6 MRCNN classification: error rate is the percentage of the negative recognition for 10,000 testing samples

#Hidden nodes	Error rate	Training time (S)
100	0.0832	198
200	0.51	232
300	0.412	250
400	0.0371	278
500	0.0365	301
600	0.032	348
700	0.0316	372
800	0.03	396
900	0.0271	428
1000	0.0279	461

hidden features are abstracted. In addition, the learning time increases certainly as the model becomes larger and larger.

In this experiment, the number of splits during MapReduce process is 20 that is determined according experience. The error rates of the classification and the computational time are shown in Table 6. From the results, it is easy to see that the accuracy is very low when there are not enough hidden nodes that are different features to represent the input patterns. However, as the number of hidden features increases, the classification error decreases. On the other hand, the computational time of the conventional learning algorithm significantly ascends. Comparatively, the computational time of the proposed distributed learning algorithm increases slowly as the number of the hidden nodes goes up. In addition, by observing the classification error rate, the proposed MRCNN learning algorithm does not lose accuracy compared with a classical learning approach. Therefore, it is safe to say that the MRCNN can significantly alleviate the computational burden of the model learning process.

Table 7 Comparison table

Metrics	Methods		
	CNN	CNN + SVM	CNN + LSTM
Accuracy (%)	90.8	93.6	95.8
Precision (%)	90.9	93.2	95.0
Recall (%)	90.2	93.4	95.7

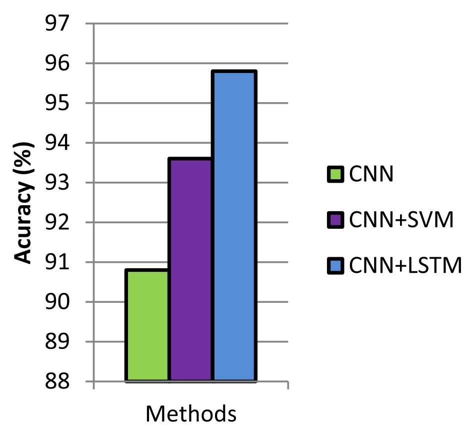


Fig. 11 Accuracy comparison

5.4 Accuracy, precision, and recall

The performance that was achieved by the proposed method was compared to that support vector mechine (SVM) (Furey et al. 2000), CNN and LSTM along with CNN. The performance comparison of the proposed and existing methods is represented in Table 7.

Figure 11 illustrates the performance of the CNN, CNN with SVM and proposed CNN with LSTM classifier in terms of accuracy metric. In x-axis method is taken and accuracy is taken as y-axis. In this proposed research work, algorithms are utilized to select fit class. It improves the accuracy rate of the classifier. From the results it can be concluded that the proposed CNN with

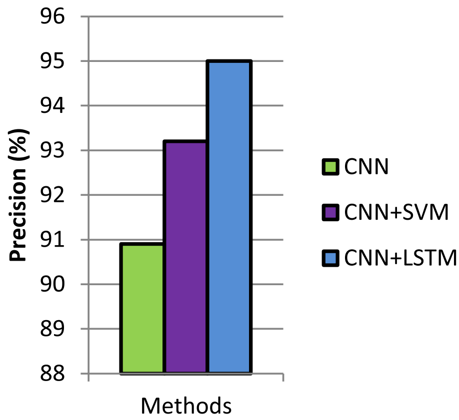


Fig. 12 Precision comparison

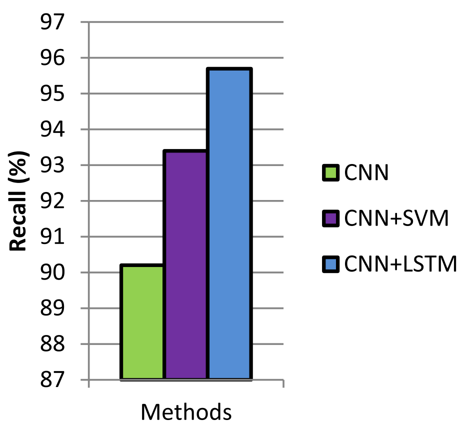


Fig. 13 Recall comparison

LSTM network attains 95.8% of accuracy whereas other methods such as CNN, CNN with SVM attains 90.8% and 93.6%, respectively. Precision performances of two approaches are compared with the CNN along with LSTM network which are shown in Fig. 12. In X-axis method is taken and precision is taken as Y-axis. The experimental results shows that CNN with LSTM classifier attains 95.0% of precision whereas other methods such as CNN, CNN with SVM attains 90.9% and 93.2%, respectively.

Figure 13 shows the recall performance of three schemes. In X-axis method is taken and recall is taken as Y-axis. As observed from the experimental results, it can be reported that CNN with LSTM network yields 95.7% of recall when CNN and CNN with SVM approach achieves 90.2% and 93.4%, respectively.

6 Conclusions

With the expansion of the number of vehicles, the need for urban traffic monitoring systems is essential. Due to the large amount of data generated in traffic control centers, we call them Big Data. In this paper, drivers' violations are classified using two types of Deep Neural Networks called Convolutional Neural Network (CNN) and Long Short Term Memory networks (LSTM). To reduce processing time and speed up the system, the MapReduce technique was used along with CNN and LSTM networks. LSTM has also been used along CNN to increase the accuracy of the design system. The evaluations of the proposed system indicated that more accurate monitoring automatically creates the power of deterrence and behavior change in drivers and it prevents drivers from committing unusual behaviors in society.

References

Aghdam HH, Heravi EJ (2017) Guide to convolutional neural networks. Springer 10:973–978

Aoyama K (1997) Next Generation Universal Traffic Management System (UTMS'21) in Japan. Intelligent Transportation System, 1997. ITSC'97, IEEE Conference on IEEE.

Arcos-García Á, Soilán M, Álvarez-García JA, Riveiro B (2017) Exploiting synergies of mobile mapping sensors and deep learning for traffic sign recognition systems. Expert Syst Appl 89:286–295

Arcos-García Á, Álvarez-García JA, Soria-Morillo LM (2018a) Deep neural network for traffic sign recognition systems: An analysis of spatial transformers and stochastic optimisation methods. Neural Netw 99:158–165

Arcos-García Á, Álvarez-García JA, Soria-Morillo LM (2018b) Evaluation of Deep Neural Networks for traffic sign detection systems. Neurocomputing 316:332–344

Asadianfam S, Shamsi M, Kenari AR (2020a) TVD-MRDL: traffic violation detection system using MapReduce-based deep learning for large-scale data. Multimed Tools Appl 80:1–28

Asadianfam S, Shamsi M, Rasouli Kenari A (2020b) Big data platform of traffic violation detection system: identifying the risky behaviors of vehicle drivers. Multimed Tools Appl 29:24645–24684

Atlas LE, Homma T, Marks RJ (1988) An artificial neural network for spatio-temporal bipolar patterns: application to phoneme classification. In: Anderson DZ (ed) Neural Information Processing Systems, pp 31–40

Cantabella M, Martínez-España R, Ayuso B, Yáñez JA, Muñoz A (2019) Analysis of student behavior in learning management systems through a Big Data framework. Future Gener Comput Syst 90:262–272

Chen M, Mao S, Liu Y (2014a) Big data: a survey. Mobile Netw Appl 19(2):171–209

Chen M, Mao S, Zhang Y, Leung VC (2014b) Big data: related technologies, challenges and future prospects. Springer, Berlin

De Mauro A, Greco M, Grimaldi M, Ritala P (2018) Human resources for Big Data professions: a systematic classification of job roles and required skill sets. Inf Process Manag 54(5):807–817

Dean J, Ghemawat S (2008) MapReduce: simplified data processing on large clusters. Commun ACM 51(1):107–113

Dvornek NC, Ventola P, Pelphrey KA, Duncan JS (2017) Identifying autism from resting-state fMRI using long short-term memory networks. International workshop on machine learning in medical imaging. Springer, Berlin

Fukushima K, Miyake S (1982) Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition. Competition and cooperation in neural nets. Springer, Berlin, pp 267–285

Furey TS, Cristianini N, Duffy N, Bednarski DW, Schummer M, Haussler D (2000) Support vector machine classification and validation of cancer tissue samples using microarray expression data. Bioinformatics 16(10):906–914

Gantz J, Reinsel D (2011) Extracting value from chaos. IDC Iview 1142(2011):1–12

Goh YM, Chua D (2009) Case-based reasoning for construction hazard identification: case representation and retrieval. J Constr Eng Manag 135(11):1181–1189

Goodfellow I, Bengio Y, Courville A (2016) Deep learning (adaptive computation and machine learning series). Adaptive Computation and Machine Learning series, MIT Press

Guo S, Ding L, Luo H, Jiang X (2016) A Big-Data-based platform of workers' behavior: observations from the field. Accid Anal Prev 93:299–309

Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780

Hubel DH, Wiesel TN (1968) Receptive fields and functional architecture of monkey striate cortex. J Physiol 195(1):215–243

Jeon S, Hong B (2016) Monte Carlo simulation-based traffic speed forecasting using historical big data. Future Gener Comput Syst 65:182–195

Kim H (2019) Multiple vehicle tracking and classification system with a convolutional neural network. J Ambient Intell Humaniz Comput. https://doi.org/10.1007/s12652-019-01429-5

Kumar CR, Anuradha R (2020) Feature selection and classification methods for vehicle tracking and detection. J Ambient Intell Human Comput. https://doi.org/10.1007/s12652-020-01824-3

Kutlu H, Avcı E (2019) A novel method for classifying liver and brain tumors using convolutional neural networks, discrete wavelet transform and long short-term memory networks. Sensors 19(9):1992

Laney D (2001) 3D data management: controlling data volume, velocity and variety. META Group Res Note 6(70):1

LeCun Y, Boser B, Denker JS, Henderson D, Howard RE, Hubbard W, Jackel LD (1989) Backpropagation applied to handwritten zip code recognition. Neural Comput 1(4):541–551

LeCun Y, Bengio Y, Hinton G (2015) Deep learning. Nature 521(7553):436–444

Lotfi E (2012) Trajectory clustering and behaviour retrieval from traffic surveillance videos. Majlesi J Multimed Process 1:1–8

Lu W (2019) Big data analytics to identify illegal construction waste dumping: A Hong Kong study. Resour Conserv Recycl 141:264–272

McLauchlan P, Beymer D, Coifman B, Mali J (1997) A real-time computer vision system for measuring traffic parameters. cvpr, IEEE.

Moghaddam AM, Ayati E (2014) Introducing a risk estimation index for drivers: a case of Iran. Saf Sci 62:90–97

Munoz-Organero M, Ruiz-Blaquez R, Sánchez-Fernández L (2018) Automatic detection of traffic lights, street crossings and urban roundabouts combining outlier detection and deep learning classification techniques based on GPS traces while driving. Comput Environ Urban Syst 68:1–8

Osman AMS (2018) A novel big data analytics framework for smart cities. Future Gener Comput Syst 91:620–633

Park SH, Jung K, Hea JK, Kim HJ (1999) Vision-based traffic surveillance system on the internet. Computational Intelligence and Multimedia Applications, 1999. ICCIMA'99. Proceedings. Third International Conference on, IEEE.

Patterson J, Gibson A (2017) Deep learning: a practitioner's approach. O'Reilly Media Inc., Newton

Rahemi Z, Ajorpaz NM, Esfahani MS, Aghajani M (2017) Sensation-seeking and factors related to dangerous driving behaviors among Iranian drivers. Personality Individ Differ 116:314–318

Rakotonirainy A, Schroeter R, Soro A (2014) Three social car visions to improve driver behaviour. Pervasive Mobile Comput 14:147–160

Rao X, Lin F, Chen Z, Zhao J (2019) Distracted driving recognition method based on deep convolutional neural network. J Ambient Intell Human Comput. https://doi.org/10.1007/s12652-019-01597-4

Schmidhuber J (2015) Deep learning in neural networks: an overview. Neural Netw 61:85–117

Secundo G, Del Vecchio P, Dumay J, Passiante G (2017) Intellectual capital in the age of Big Data: establishing a research agenda. J Intellect Cap 18(2):242–261

Shvachko K, Kuang H, Radia S, Chansler R (2010). The hadoop distributed file system. Mass storage systems and technologies (MSST), 2010 IEEE 26th symposium on, IEEE.

Tao D, Zhang R, Qu X (2017) The role of personality traits and driving experience in self-reported risky driving behaviors and accident risk among Chinese drivers. Accid Anal Prev 99:228–235

Wallach I, Dzamba M, Heifets A (2015) AtomNet: a deep convolutional neural network for bioactivity prediction in structure-based drug discovery. arXiv preprint http://arxiv.org/abs/1510.02855

White T (2012) Hadoop: the definitive guide. O'Reilly Media Inc, Newton

Yin L, Cheng Q, Wang Z, Shao Z (2015) 'Big data'for pedestrian volume: Exploring the use of Google Street View images for pedestrian counts. Appl Geogr 63:337–345

Yu Y, Li J, Wen C, Guan H, Luo H, Wang C (2016) Bag-of-visual-phrases and hierarchical deep models for traffic sign detection and recognition in mobile laser scanning data. ISPRS J Photogramm Remote Sens 113:106–123