



# Trajectory-level fog detection based on in-vehicle video camera with TensorFlow deep learning utilizing SHRP2 naturalistic driving data



Md Nasim Khan\*, Mohamed M. Ahmed

*University of Wyoming, Department of Civil & Architectural Engineering, 1000 E University Ave, Dept. 3295, Laramie, WY 82071, United States*

## ARTICLE INFO

**Keywords:**  
 Foggy weather  
 Machine Learning  
 Deep Learning  
 Image Classification  
 TensorFlow  
 Deep Neural Network  
 Recurrent Neural Network  
 Long Short-Term Memory  
 Convolutional Neural Network  
 Advanced Driver Assistance Systems  
 Advanced Travel Information Systems  
 Variable Speed Limit  
 Mobile Weather Sensors

## ABSTRACT

Providing drivers with real-time weather information and driving assistance during adverse weather, including fog, is crucial for safe driving. The primary focus of this study was to develop an affordable in-vehicle fog detection method, which will provide accurate trajectory-level weather information in real-time. The study used the SHRP2 Naturalistic Driving Study (NDS) video data and utilized several promising Deep Learning techniques, including Deep Neural Network (DNN), Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM), and Convolutional Neural Network (CNN). Python programming on the TensorFlow Machine Learning library has been used for training the Deep Learning models. The analysis was done on a dataset consisted of three weather conditions, including clear, distant fog and near fog. During the training process, two optimizers, including Adam and Gradient Descent, have been used. While the overall prediction accuracy of the DNN, RNN, LSTM, and CNN using the Gradient Descent optimizer were found to be around 85 %, 77 %, 84 %, and 97 %, respectively; much improved overall prediction accuracy of 88 %, 91 %, 93 %, and 98 % for the DNN, RNN, LSTM, and CNN, respectively, were observed considering the Adam optimizer. The proposed fog detection method requires only a single video camera to detect weather conditions, and therefore, can be an inexpensive option to be fitted in maintenance vehicles to collect trajectory-level weather information in real-time for expanding as well as updating weather-based Variable Speed Limit (VSL) systems and Advanced Traveler Information Systems (ATIS).

## 1. Introduction

Foggy weather condition is one of the most frequent causes of traffic accidents due to its adverse impact on visibility. Driver performance can be drastically affected by foggy weather conditions because they may not be able to accurately assess the visibility distance as well as electing safe speeds for the foggy weather condition (Khan et al., 2018). According to the Federal Highway Administration (FHWA), around 10,448 injury crashes and 495 fatal crashes occur every year in the U.S. due to the presence of fog (FHWA, 2019). Many previous studies have concluded that foggy weather conditions have a significant negative impact on driver behavior and performance as well as one of the major causes of weather-related motor vehicle crashes (Abdel-Aty et al., 2011; Ahmed et al., 2014; Das and Ahmed, 2019; Wu et al., 2018). Therefore, proper assessment of road visibility conditions and providing drivers with appropriate warnings and speed limits are crucial for safe driving, especially in adverse weather, including fog.

Considering the negative impact of reduced visibility on traffic safety, many studies have thoroughly investigated the relationship

between real-time crash likelihood and visibility using weather station data. A study by Ahmed et al. (2014) examined the feasibility of using weather information collected from nearby airports for the assessment of road crashes in real-time on highways with frequent fog problem using Bayesian logistic regression and found that the likelihood of crash occurrence might increase with the reduction in visibility. Based on information from weather stations and real-time traffic data, the study in Wu et al. (2018) concluded that fog conditions increase the likelihood of crashes, especially in congested traffic conditions. Another study (Yu et al., 2015) also used weather station data to investigate the effect of visibility on crash risk for a mountainous freeway and concluded that poor visibility conditions would increase the probability of crashes. Many other studies (Yu et al., 2013; Usman et al., 2010) using weather station data also found similar results.

Fog can cause a sudden reduction in roadway visibility and dramatic changes in driving conditions due to its variable nature as well as the small area over which it usually forms. This unpredictability of fog formation makes it one of the major causes of motor vehicle crashes. Abrupt reduction in visibility due to fog often leads to pileup and more

\* Corresponding author.

E-mail addresses: [mkhan7@uwyo.edu](mailto:mkhan7@uwyo.edu) (M.N. Khan), [mahmed@uwyo.edu](mailto:mahmed@uwyo.edu) (M.M. Ahmed).

severe crashes. As an example, a 19-vehicle pileup crashes occurred on Interstate 75, Florida, due to sudden drop in visibility caused by thick fog, killing ten and injuring more than twenty people (Hassan and Abdel-aty, 2013). Despite these adverse effects on roadway safety, the sudden local fog formation sometimes cannot be detected by weather stations due to the fact that weather stations are location-specific and weather sensors are mostly mounted at higher elevations that may not necessarily represent visibility at road surface level (Khan and Ahmed, 2019). Therefore, in-vehicle sensors, including video cameras, are more promising for better representative real-time visibility determination.

Different approaches to video camera-based visibility estimation can be found in the literature. For example, Pomerleau, 1997 introduced a system based on the contrast attenuation of the lane markings at a different distance in front of the vehicle. He used Rapidly Adapting Lateral Position Handler (RALPH) method to find and track roadway features, including lane markings, road/shoulder boundaries, tracks left by other vehicles, and pavement discoloration. A study by Hautière et al. (2006a) used a disparity map for visibility determination and tested their system in sunny, foggy, and dense foggy weather conditions. The average visibility distance was found to be approximately 250 m, 75 m, and 30 m in sunny weather, foggy weather, and dense foggy weather, respectively. Another study (Hautière et al., 2006b) by the same authors applied Koschmieder's law ( $V_{met} = \frac{3}{k}$ ) to estimate visibility distance. Here,  $V_{met}$  is the meteorological visibility distance proposed by the International Commission on Illumination (CIE), and  $k$  is the extinction coefficient of the atmosphere. First, the Road Surface Luminance Curve (RSLC) was extracted using roadway images taken from an in-vehicle video camera. Subsequently, a relationship between the RSLC inflection point and extinction coefficient was derived, in which  $k$  could be estimated. They showed that the RSLC model should consider the effects of non-horizontal vision. All RSLC-based approaches can work on just a single frame. However, they depend strongly on a specific road scenario that does not allow for objects blocking the view to the horizon. The recent approaches to determine visibility from video cameras are mainly based on machine vision techniques. Bronte et al. (2009) proposed a real-time fog detection system using an onboard low-cost black and white camera. Their system is based on two clues; estimation of the visibility distance, which is calculated from the camera projection equations, and the blurring due to the fog. The polarized-based method, which requires two or more images of the same scene taken with different degrees of polarization, was proposed by Schechner et al. (2001) and Shwartz et al. (2006). The problem of the polarized based method is that it requires multiple images of the same scene. To overcome this problem, Tan (2008) developed an automated fog detection method that only requires a single input image. Pavlić et al. (2012) developed an in-vehicle fog detection system based on global image descriptors and a classification procedure. Although the granularity of their system could not identify different levels of fog, they achieved a classification accuracy of 96 % and 93 % for clear and foggy conditions, respectively. Zhang et al. (2015) proposed a technique using Support Vector Machine (SVM) and nine different features as individual predictors and achieved a maximum accuracy of 78.5 % (Zhang et al., 2015).

Most of the current in-vehicle fog detection systems are based on a consistent object (lane marking, signs, the horizon, etc.) in front of the vehicle to operate correctly. These fog detection methods are not dependable in everyday situations because the road marking, signs, and the horizon can be obstructed by other vehicles, especially in congested traffic conditions. Image-based fog detection technique using deep learning may overcome these problems as it only uses the global features of images. However, the potential of deep learning for real-time road surface visibility identification is not notably researched. This study proposes some unique techniques to enhance the reliability of real-time detection of road surface visibility from video data collected from the SHRP2 Naturalistic Driving Study (NDS) dataset. The study is based on several promising deep learning techniques, such as Deep

Neural Network (DNN), Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM), and Convolutional Neural Network (CNN). Python programming on the TensorFlow machine learning library has been used for training the deep learning models.

In recent years, various fields of engineering are extensively using deep learning for classification, detection, and pattern recognition purposes due to its advantages over traditional statistical and machine learning techniques in terms of accuracy, especially when trained with Big Data. In the transportation field, many studies have used deep learning to detect and predict crashes, conflicts, vehicles, traffic flow, speed, travel time, and driver distraction. A recent study (Huang et al., 2020) used several variations of deep learning to detect crash occurrence and predict crash risk using speed, volume, and occupancy data; and compared the results with traditional machine learning techniques. The study concluded that deep learning models performed better in terms of accuracy for crash detection compared to state-of-the-art shallow models. Another study (Bao et al., 2019) proposed a deep learning architecture, which they called spatiotemporal convolutional long short-term memory network (STCL-Net), to predict crashes. They argued that the proposed deep model could leverage both the spatial and temporal dependencies in the explanatory variables. The results indicated that the proposed deep models outperformed the traditional statistical as well as state-of-the-art machine-learning models based on several evaluation criteria, including prediction accuracy and false alarm rate. A study by Formosa et al. (2020) utilized deep learning to leverage large imbalanced data for predicting real-time traffic conflicts considering several influencing factors including, speed variance, traffic density, speed, and weather conditions. The results indicated that the proposed conflict detection techniques could be effectively integrated to improve ADAS, CVs, and AVs. The study by Lv et al. (2015) proposed a novel traffic flow prediction method with Big traffic data using deep learning. The study claimed that unlike other models, the proposed model could effectively uncover latent nonlinear spatial and temporal associations from traffic data. They concluded that the deep model is superior compared to traditional machine learning models in terms of prediction performance. Another study by Napoletano and Schettini (2018) proposed a method capable of automatically detect driver distraction using deep learning and convolutional neural networks using video cameras mounted inside vehicles pointing at drivers. Many similar studies have also used deep learning to detect driver distraction automatically (Alotaibi and Alotaibi, 2019; Hashemi et al., 2020). Deep learning has also been used for travel time estimation (Tang et al., 2019), speed prediction (Li et al., 2019), traffic signal control (Xu et al., 2020; Jeon et al., 2018), and vehicle defect detection (Krummenacher et al., 2018). Although many studies have used deep learning to detect weather conditions using climate data (Prabhat et al., 2017) or image data from stationary cameras (Ibrahim and Haworth, 2019), the use of deep learning to detect trajectory-level weather at road surface level using naturalistic data is limited. Considering this research gap in mind, in this study, a novel approach for detecting road surface level fog in real-time based on deep neural networking techniques has been proposed.

It is worth noting that the negative impact of adverse weather on road surface visibility is more noticeable in foggy weather compared to other weather conditions (Codling, 1971). In addition, fog and snow do not usually occur at the same time of the year in some states such as Wyoming. Also, snowfall is a very rare environmental condition in the southern part of the U.S., including Florida and Texas (Stone, 1936). Considering the reasons mentioned above, this study only focused on fog detection in order to eliminate any possible reduction in detection accuracy due to the presence of other weather conditions in the model.

## 2. Data reduction

The SHRP2 NDS video data used in this study were acquired from the Virginia Tech Transportation Institute (VTI). The SHRP2 is the most extensive study on naturalistic driving behavior until now in the U.S.

**Table 1**

Criteria for Classifying Weather Conditions during Image Annotation.

Object	Visibility		
	Clear	Distant Fog	Near Fog
Horizon	Visible	Undefinable	Undefinable
Road markings	Visible	Visible	No/only few road markings are visible
Road signs	Clearly readable	Readable to some extent	Unreadable
Road surroundings (e.g., guardrails, delineators New Jersey barriers, etc.) and ambient traffic properly	Visible	Visible	Not clearly visible

Many previous studies have used the data from this comprehensive study to investigate driver behavior and performance under different weather and traffic conditions in an attempt to improve the safety of the roadways (Ahmed et al., 2017; Ali et al., 2019; Das et al., 2020a, 2020b, 2019; Das and Ahmed, 2019). Participants' vehicles were instrumented with a proprietary Data Acquisition System (DAS), which was developed by the VTTI solely for the SHRP2 program. The DAS includes forward radar; four video cameras, including one forward-facing color wide-angle camera; accelerometers; vehicle network information; Geographic Positioning System (GPS); onboard computer vision lane tracking, plus other computer vision algorithms; and data storage capability (Hankey et al., 2016; Smadi et al., 2015). However, for this study, only the data of the forward-facing color video camera were used.

The data reduction started with the acquisition of the video data of interest from the massive SHRP2 NDS dataset. The SHRP2 NDS collected more than two petabytes of data in different weather conditions (Hankey et al. 2016); therefore, identifying trips of interest was extremely challenging. To overcome this problem, the research team developed two methods to effectively extract video data of trips that occurred in foggy weather (Ahmed et al., 2018). The first method utilized the weather station data archived in the National Climate Data Center (NCDC). To identify the zones affected by foggy weather, a circular area of five nautical miles around each station was considered (Ahmed et al., 2014). Subsequently, NDS trips were requested based on the daily weather information to identify all trips impacted by foggy weather. The daily weather information was used only to identify the potential trips that occurred in foggy weather conditions. It is worth mentioning that hourly weather information cannot be used considering the fact that the exact time and date of trips are considered Personally Identifiable Information (PII) and hence were not provided. PII refers to any information that has the potential to identify a specific individual. The second method used weather-related crash data to identify potential locations of trips that occurred in foggy weather. By using these methods, video data of trips occurring in foggy and clear weather were acquired from the VTTI data enclave.

Once the video data of adverse and clear weather were collected, all the videos were observed manually to eliminate NDS trips that did not occur in foggy weather. A total of 217 video recordings of naturalistic trips occurred in foggy weather, and their matched 430 video recordings of trips occurred in clear weather were considered for further analysis. It is worth mentioning that the initial matching was conducted by VTTI; however, to ensure more accurate matching, the longitude and

latitude of the trips were imported into the ArcGIS software and subsequently, the non-matching segments were identified and eliminated from the videos. After that, twelve images were extracted per minute from all the videos, which resulted in a total of more than 20,000 images in clear and foggy weather conditions. Note that for detection of roadway features, such as signs, signals, and work zones, higher sampling rate is required; however, for weather detection, 12 frames per minute (e.g., one image in every 5 s) is a reasonable resolution, considering the fact that weather will most likely be constant within 5 s. For training Deep Learning models, correct annotation of images is more important than a higher sampling rate (Khan and Ahmed, 2019).

Once the extraction of images has been completed, the images were annotated manually and grouped into three categories: clear, distant fog, and near fog. It is worth noting that the classification of fog is not consistent in the literature. The National Oceanic and Atmospheric Administration (NOAA) classified fog into two categories (Mueller, 1959). They classified fog as near if the visibility distance falls below 0.25 miles and light if the visibility distance is between 0.3 miles to 6 miles. In 1992, the South Carolina Department of Transportation (SCDOT) developed a low visibility warning system, where they defined fog as dense if the visibility falls below 300 feet and light if the visibility ranges between 300 feet to 900 feet (Murphy et al., 2012). However, for this study, fog was classified into two categories (i.e., distant fog and near fog) using qualitative-based measures extracted from the NDS videos. The fog was classified based on the visibility of road markings, readability of road signs, roadside surroundings (delineators, guardrail, New Jersey barriers, and so forth) and the horizon. It is worth mentioning that these objects were only used during the manual annotation process because the human cannot perceive individual pixels and change in pixel intensity resulted from fog. However, the proposed neural network models use global features of the images and are not dependent on any arbitrary object and hence, can be applied in everyday scenarios, including congested traffic conditions. In clear weather, it is more likely to have a sudden change in pixel intensity, whereas, in foggy weather, more uniform pixel intensity is expected. The deep neural network models can uncover these latent relationships between change in pixel intensity of the images and weather conditions. The criteria for classifying weather conditions during the manual image annotation are listed in Table 1, and sample images of weather conditions are shown in Figs. 1–3.

It is worth mentioning that the initial image annotation might have some potential error due to subjectivity since all the images were



**Fig. 1.** Sample images of Clear Weather. a) and b) The horizon, road markings, roadside surroundings are visible; Signs are readable.



**Fig. 2.** Sample images of Distance Fog a) Road markings are visible, signs are readable, surroundings and traffic can be seen to some extent, the horizon cannot be seen clearly. b) Road markings are visible, speed limit signs are readable, surroundings and traffic can be seen, the horizon cannot be seen clearly.

annotated by a single observer. Therefore, in order to validate the consistency of image annotation, 20 % of the data were randomly selected and provided to three external observers. Subsequently, based on the weighted results from the observers, the final label of each image was determined and compared with the original label, which revealed that the accuracy of the initial annotation was 99.7 %, 99 %, and 96 %, for clear, distant fog, and near fog images, respectively. In order to eliminate subjectivity and any potential bias in identifying weather, comprehensive training, and a detailed description of each weather condition with sample images were provided to the external observers. However, data from the initial image annotation was used to develop the fog detection models since the overall consistency of the initial annotation is more than 98 %.

Once the manual image annotation was completed, the images were grouped into two datasets: training dataset and testing dataset. The training dataset consisted of 8000 clear weather images, 6800 distant fog images, and 1200 near fog images, which resulted in a total of 16,000 images and was equivalent to around 1,333 min of video data. In addition, the testing dataset consisted of 2000 clear weather images, 1700 distant fog images, and 300 near fog images, which resulted in a total of 4000 images and was equivalent to around 333 min of video data. As near fog is a relatively rare environment condition, the number of near fog images were respectively less compared to other categories. **Table 2** summarizes the image datasets used in this study.

### 3. Methodology

In this study, neural network (NN) method was adopted to develop the fog detection method using the annotated image dataset. NN is a machine learning approach which has been extensively used in various field of engineering for image classification, pattern recognition, and text categorization (Sebastiani, 2002; Abadi et al., 2016) as well as in many traffic safety-related research (Jacobé de Naurois et al., 2019; Wang et al., 2019). Different variations of the traditional NN were developed to serve specific purposes, e.g., RNN for time-series data analysis and CNN for image classification. This study explored the potential of the traditional NN, as well as the RNN, LSTM, and CNN, for fog detection. The NN models can be developed using various machine learning libraries under different programming platforms such as Python, Java, and C++. However, for this study, Python programming

**Table 2**  
Summary Statistics of Image Datasets.

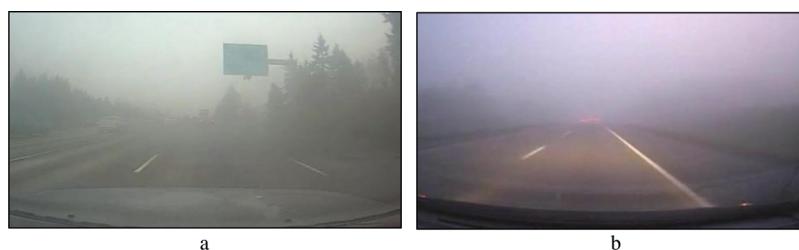
	Weather	Number of Images	Equivalent video Duration (min)
Training Dataset	Clear	8000	666.67
	Distant Fog	6800	566.67
	Near Fog	1200	100
	<b>Total</b>	<b>16,000</b>	<b>1333.33</b>
Testing Dataset	Clear	2000	166.67
	Distant Fog	1700	141.67
	Near Fog	300	25.00
	<b>Total</b>	<b>4000</b>	<b>333.33</b>

on TensorFlow machine learning library was used.

TensorFlow is a fast, flexible, and scalable open-source Machine Learning library that can be used to implement a wide variety of machine learning algorithms. TensorFlow was developed for conducting machine learning and deep learning research by the researchers working on Google's Machine Intelligence research organization (Google, 2018). The primary building block of any TensorFlow network is tensor, which is defined as a multi-dimensional array. The nodes inside a TensorFlow network can only receive data in a tensor form. Therefore, all the input data need to be converted into tensor before feeding them into the network.

The basic unit of a NN is neuron, also called node, which receives input from some other neurons, or from an external source, and computes an output. Each input has an associated weight ( $W$ ), which is assigned based on its relative importance to the other inputs. The node applies a function ( $a$ ), called the activation function, to the weighted sum of its inputs. The activation function receives a single number and performs a specific mathematical operation on it. The activation function either returns one (neuron triggered) or zero (neuron not triggered) depending on the computations performed over the weights and biases. The neuron shown in Fig. 4 has two numeric inputs ( $X_1$ ) and ( $X_2$ ) with weights ( $W_1$ ) and ( $W_2$ ), respectively. Additionally, there is another input, known as bias, with weight ( $B$ ) associated with it. Inside the neuron, the weighted sum ( $S$ ) of all the inputs have been calculated before passing it through an activation function ( $a$ ) to generate the output for the next node.

Another important parameter of a NN is cost, which captures the



**Fig. 3.** Sample images of Near Fog: a) Only one road marking is visible, signs are unreadable, surroundings, and the horizon cannot be seen properly. b) Few road markings are visible; surroundings, traffic, and the horizon cannot be seen clearly.

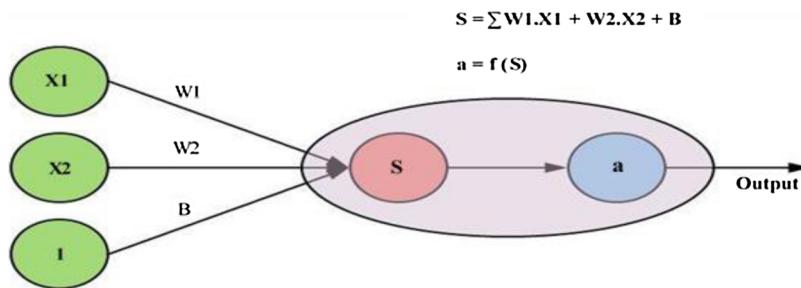


Fig. 4. A Single Neuron.

difference between the predicted and the true class. The cost, also known as loss, is an overall measure of the performance of the trained model and is represented by a single value. The cost is a function of weights ( $W$ ), biases ( $B$ ), inputs of the training sample ( $I_T$ ) and the desired outputs of the training sample ( $O_T$ ) (Karn, 2018). The general form of a cost function can be described using Eq. (1).

$$\text{Cost, } C = f(W, B, I_T, O_T) \quad (1)$$

A cost function must satisfy two properties: first, it must represent the average deviation of the predicted class from the true class; second, it should be independent of any activation value of the NN except the output values. Although, various cost functions including, quadratic, cross-entropy, exponential, Hellinger distance, Kullback-leibler divergence, and Itakura-satio are currently being used to determine the deviation of the predicted class from the true class, the most commonly used cost function for a NN model is cross-entropy (Goodfellow et al., 2016). The NN models proposed in this study also utilized the cross-entropy as a cost function. The cost using cross-entropy can be defined by the following equation.

$$\text{Cost, } C = - \sum_i [y_i \ln a_i + (1 - y_i) \ln (1 - a_i)] \quad (2)$$

Where  $y_i$  is the predicted probability value for the class  $i$  and  $a_i$  is the true probability for that class (TensorFlow, 2018).

In a NN model, the cost is minimized at every step using an optimizer, which modifies the weights and biases at each iteration and feeds the value to the next iteration. The two most commonly used optimizers in a NN model are Gradient Descent optimizer and Adaptive Moment Estimation (Adam) optimizer. The Gradient Descent minimizes the cost function by changing the model parameters, i.e., weight ( $W$ ) and bias ( $B$ ), in the opposite direction of the gradient of the cost function with respect to the parameters. The Gradient Descent can be described by the following equation.

$$P_t = P_{t-1} - \lambda \nabla f(P_{t-1}) \quad (3)$$

Where  $P_t$  represents the value of the parameters (e.g.,  $W$  and  $B$ ) for the next step,  $P_{t-1}$  represent the current value of the parameters,  $\lambda$  is a weighting factor, and the gradient term  $\nabla f(P_{t-1})$  is the direction of the steepest descent (Donges, 2018). In order for the Gradient Descent to reach the optimum values of the parameter, an appropriate learning rate need to be selected. A large learning rate may not provide optimum value because it will bounce back and forth between the convex function of the Gradient Descent. Conversely, a small learning rate will significantly increase the training time. Therefore, the learning rate should be selected in such a way that the parameters achieve optimum values within the least possible time.

While Gradient Descent maintains a constant learning rate for all the parameter updates, the Adam optimizer, which was first proposed by Kingma et al. (2014), computes individual adaptive learning rates for each parameter. Adam is an extension of the stochastic gradient descent (SGD) algorithm and was developed mainly for training deep neural networks. Adam optimizer leverages the advantages of two other extensions of SGD: Adaptive Gradient Algorithm (AdaGrad), which

adapts the learning rate of the parameters based on their update frequency during training and works well with sparse gradients (Duchi, 2011); and Root Mean Square Propagation (RMSProp), which adjusts the parameter learning rate based on the magnitude of the recent gradients and works well in non-stationary setting (Tieleman and Hinton, 2012).

For optimization, Adam first computes the gradient (1st-order moment) and element-wise squared gradient (2nd-order moment). These moments can be described using the following equations:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \quad (4)$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \quad (5)$$

Here  $m_t$  and  $v_t$  are the moving averages of the gradient and squared gradient at time step  $t$ , respectively;  $g_t$  is the gradient of the current mini-batch; and  $\beta_1$  and  $\beta_2$  are the hyperparameters. It is worth mentioning that the moments (i.e.  $m_t$  and  $v_t$ ) are usually biased towards zero during the initial time steps and can be corrected using the following equations:

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t} \quad (6)$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t} \quad (7)$$

Here  $\hat{m}_t$  and  $\hat{v}_t$  bias-corrected estimators for the first and second moments, respectively, which are then used to update the weights of the model using the following equation:

$$w_t = w_{t-1} - \eta \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} \quad (8)$$

Here  $w_t$  and  $w_{t-1}$  are the weights of the model at timestep  $t$  and  $t - 1$ ,  $\eta$  is the step size, and  $\epsilon$  is an empirical term introduced to avoid errors due to divisions by zero (Ruder, 2016).

Architecturally, a NN graph consists of three layers: input layer that receives inputs such as images and converts them to a mathematical form so that the neuron can interpret; one or multiple hidden layers that perform necessary numerical computations on the input data from the previous layer; and finally the output layer which is responsible for transferring information from the network to the outside world. A NN is termed as Deep Neural Network (DNN) if it has more than one hidden layer. In a NN model, the order of computation needs to be determined. Although the value of the nodes can be calculated separately, the common practice of calculating node values is to arrange the nodes into layers. This technique is called feedforward. A feedforward network takes the inputs in the lowest layer (input layer). Subsequently, the higher layers (e.g., hidden layers) are calculated until the output is generated at the topmost layer. Fig. 5 provides a graphical representation of a multilayer feedforward DNN consists of three inputs, three hidden layers, and one output layer.

The DNN used in this study consists of one input layer, three hidden layers with five hundred nodes in each layer, and one output layer. The training of the DNN model started with feeding the training data into

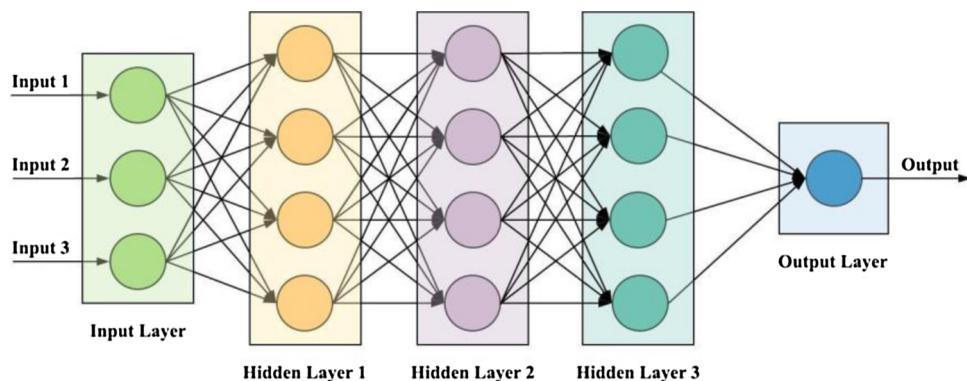


Fig. 5. Fully Connected Deep Neural Network (DNN).

the network. As mentioned earlier, the training dataset consisted of 16,000 annotated images in three different weather conditions: clear, distant fog, and near fog. The sigmoid function was used as the activation function. For calculating the cost, Cross-entropy has been used, which was calculated using Eq. (2). Two optimizers, including Adam and Gradient Descent, were used to optimize the cost function.

One of the shortcomings of the DNN is that it does not consider the sequence of data, which is essential for time series and video data. Therefore, to leverage the sequential information, a modification of DNN, known as Recurrent Neural Network (RNN), was also used in this study. The RNN treated data as a sequence using cyclic connections. The RNN stores information from the previous timestamp and uses the information as inputs to the network to compute the predictions at the current time step. At time  $t$ , if hidden nodes with recurrent connection collect inputs from the current data point  $x_t$ , and from the hidden node values  $h_{t-1}$  of the previous step; the output  $y_t$  at time  $t$  can be described using the following equations.

$$y_t = \text{softmax}(W_{yh}h_t + b_y) \quad (4)$$

$$h_t = \sigma(W_{hx}x_t + W_{hh}h_{t-1} + b_h) \quad (5)$$

Where  $\sigma$  is the activation function;  $h_t$  is the hidden node value at the time  $t$ ;  $W_{hx}$  is the conventional weight matrix based on the current input;  $W_{hh}$  is the recurrent weight matrix based on the previous hidden states;  $W_{yh}$  is the weight matrix based on hidden state and output; and  $b_h$  and  $b_y$  are the bias parameters (Lipton et al., 2015).

One of the major limitations of the RNN is that it suffers from the vanishing and exploding gradient problem, especially when optimized over numerous time steps. The vanishing gradient problem occurs when the gradient of the cost function moves toward zero with the addition of more layers to the networks, which makes the network challenging to train. On the other hand, the exploding gradient problem occurs when the gradients with values higher than one grow exponentially through the network layers, making learning unstable (Goodfellow et al., 2016). Long Short-Term Memory (LSTM) can overcome these problems by replacing the hidden layers with a memory cell that can act as intermediate storage (Lipton et al., 2015). Each memory cell consists of memory blocks with three multiplicative nodes named input, output, and forget gates, as shown in Fig. 6.

The LSTM produces an output probability vector ( $h_t$ ) from input sequence vectors ( $x_t$ ) by analyzing the network unit activations using the following equations.

$$i_t = \sigma(W_{ix}X_t + W_{ih}h_{t-1} + W_{ic}c_{t-1} + b_i) \quad (6)$$

$$f_t = \sigma(W_{fx}X_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f) \quad (7)$$

$$o_t = \sigma(W_{ox}X_t + W_{oh}h_{t-1} + W_{oc}c_t + b_o) \quad (8)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tanh(W_{cx}X_t + W_{ch}h_{t-1} + b_c) \quad (9)$$

$$h_t = o_t \odot \tanh(c_t) \quad (10)$$

$$y_t = W_{yh}h_{t-1} + b_y \quad (11)$$

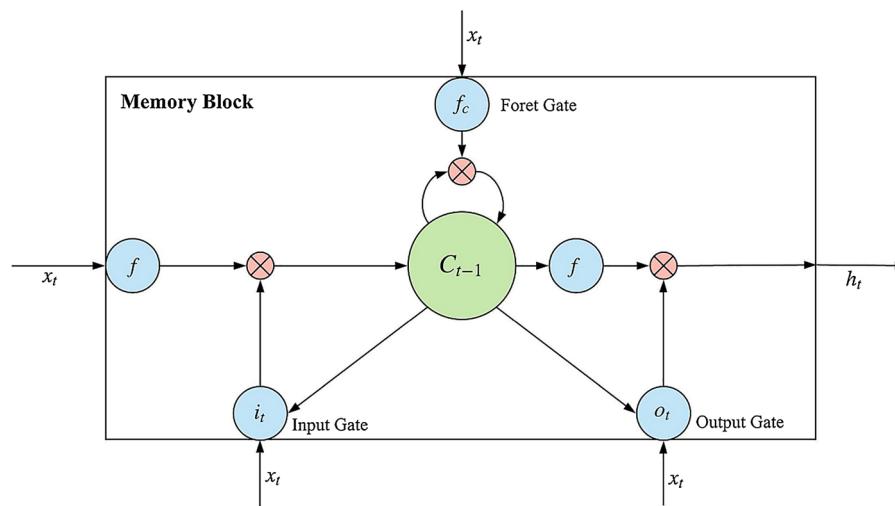
Here,  $i$ ,  $f$ ,  $o$ , and  $c$  represents the input gate, output gate, forget gate, and activation vectors;  $W_{ix}$ ,  $W_{fx}$ ,  $W_{ox}$  are the weight matrix of the input, forget, and output gates;  $h_t$  and  $h_{t-1}$  are the hidden vector at time  $t$  and  $t-1$ ;  $\sigma$  is the logistic sigmoid function; and  $\odot$  represents the elementwise product of the vectors (Li et al., 2020).

Another improvement of the DNN is the Convolutional Neural Network (CNN), which is developed primarily for image classification. A CNN consists of input layers, output layers, and many layers in between. The in-between layers can be categorized into two types of layers: feature detection layers and classification layers. The feature detection layer can perform three types of operation in the data, including convolution, Rectified Linear Unit (ReLU), and pooling. Convolution operation can activate certain features from the images by passing them through a set of convolutional filters. In this study, five convolutional layers were used during the training process. The first convolutional layer took the images as input and applied 32 filters, each with a height and width of 5 pixels. The second and the third convolutional layer applied 32 and 64 filters of the same size, respectively. For the next two convolutional layers, the same number of filters with the same filter size was applied. After each convolution layer, a ReLU layer was used to perform a threshold operation on each element of the input. ReLU layer maps negative values to zero to ensure faster and more accurate training. After each ReLU layer, a max-pooling layer was applied with pooling regions of  $3 \times 3$  pixels. Pooling simplifies the output by performing nonlinear down-sampling to reduce the number of parameters that the network needs to learn.

After feature extraction, the architecture of a CNN moved to classification. The next layer was a Fully Connected (FC) layer that provided a vector of three dimensions. Finally, the image dataset was passed into a SoftMax layer, which was the final layer of the CNN. The architecture of CNN used in this study is illustrated in Fig. 7.

#### 4. Results and discussions

In order to select the best performing model, the hyperparameters of all the models were updated by carefully observing the training progress and the validation results using TensorBoard. TensorBoard is a visualization toolkit inside the TensorFlow library for machine learning experimentation (Google, 2020). Fig. 7a illustrates the variation in accuracy and cost over the training steps. Note that, in order to better visualize the training progress, TensorBoard smoothed the curves using an infinite impulse response (IIR) filter, as can be seen in Fig. 8. While the accuracy of the DNN model using Adam optimizer gradually increased from 0.52 at the first step to 0.89 at the final step, the accuracy of the DNN model using Gradient Descent optimizer increased from 0.50 to 0.87. However, the superiority of Adam optimizer is only seen after about 30 training steps. Considering the cost, the DNN using the Gradient Descent optimizer provided a continuous decrease until it



**Fig. 6.** Architecture of LSTM Unit (Graves and Abdel-rahman Mohamed, 2013).

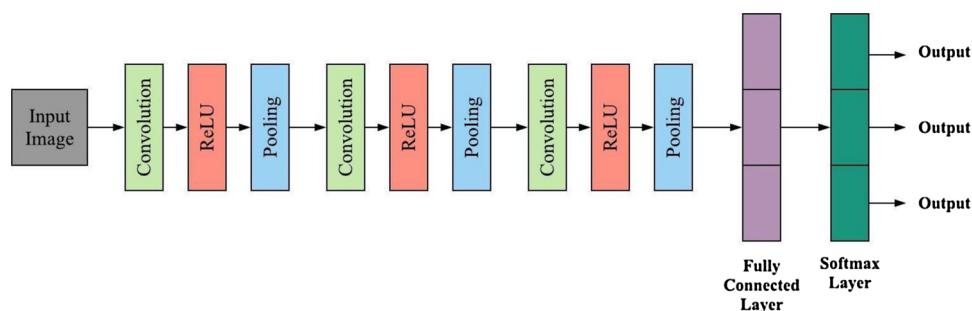
reached a minimum value. On the contrary, the cost of the DNN using Adam optimizer increased during the initial steps but eventually reached a minimum value at the final step. During the training process, the epoch, also called the training step, was set to 50, considering the fact that the accuracy of the model became almost constant, and no significant improvement was observed after 50 steps.

Fig. 8b illustrates the variation of accuracy and cost of the RNN Models in every training step. Although the RNN model using the Adam optimizer provided excellent accuracy of 0.91 at the final step of the training, the RNN model using Gradient Descent performed poorly than the Adam optimizer with a final accuracy of 0.77. The cost was also found to be much lower at every step of the training process for the RNN model trained with Adam optimizer compared to the RNN model trained with Gradient Descent, as can be seen from 5b. Moreover, no significant improvement in detection accuracy was found after 500 training steps. Fig. 8c illustrates the training progress of the LSTM model in terms of accuracy and cost. It was observed that after 500 training steps, the increase in accuracy and the decrease in cost became insignificant. Similar to other NN models, the Adam optimizer produced better results compared to the Gradient Descent optimizer with an overall validation accuracy of around 93 % at the final steps.

Considering the CNN models, the Adam optimizer performed marginally better compared to Gradient Descent optimizer, as can be seen from Fig. 8d. The accuracy of the CNN model using Adam optimizer was about 0.78 at the first step, which increased gradually and reached an accuracy of 0.97 at the last step. The CNN model using the Gradient Descent also produced similar results with an accuracy of 0.96 at the final step. The cost of the CNN model using Adam optimizer decreased gradually until it reached a minimum value. A similar trend was also observed for the CNN model using Gradient Descent optimizer, as can be seen from Fig. 8c. Note that the CNN models required a significantly

higher number of training steps in order to reach the maximum accuracy compared to the other models.

Once training of the different NN models was completed, the performance of the models was evaluated using a test dataset, which consisted of 4000 images in different weather conditions. Note that these images were not used in the training nor validation steps. The overall detection accuracy of the trained DNN, RNN, LSTM, and CNN model was found to be 85.1 %, 77.4 %, 84.2 %, and 97.3 %, respectively, using Gradient Descent optimizer, as shown in Table 3. The highest true positive (TP) rate using DNN was found for the clear image group, where 95 % of the images were correctly classified. On the other hand, the lowest TP rate was found for the distant fog image group where 73.7 % of the images were correctly classified. Considering the RNN model, the highest TP and the lowest false negative (FN) rate were found for the clear image group, where 9.5 % of the clear images were incorrectly classified to other conditions. Similarly, the TP rate of the distant fog image group was found to be 73.2 %, meaning 73.2 % of the distant fog images were correctly classified. As expected, the performance of the LSTM model using Gradient Descent optimizer was significantly better than the RNN model with an overall detection accuracy of 84.2 % compared to 77.4 % using RNN. More specifically, 84.6 %, 86.7 %, and 67.9 % of the clear, distance fog, and near fog images, respectively, were correctly classified. The lowest FN rate was found for the distance fog image group with only 13.3 % misclassification. The detection accuracy of the near fog image group for the LSTM model using Gradient Descent optimizer was lower compared to the CNN model; however, it was significantly higher compared to DNN and RNN model. One possible reason is that the sample size of near fog images was relatively smaller compared to other weather conditions since near fog is a rare environmental condition. More balanced sample size with an equal number of images from each category might provide improved



**Fig. 7.** Representation of a Convolutional Neural Network (CNN).

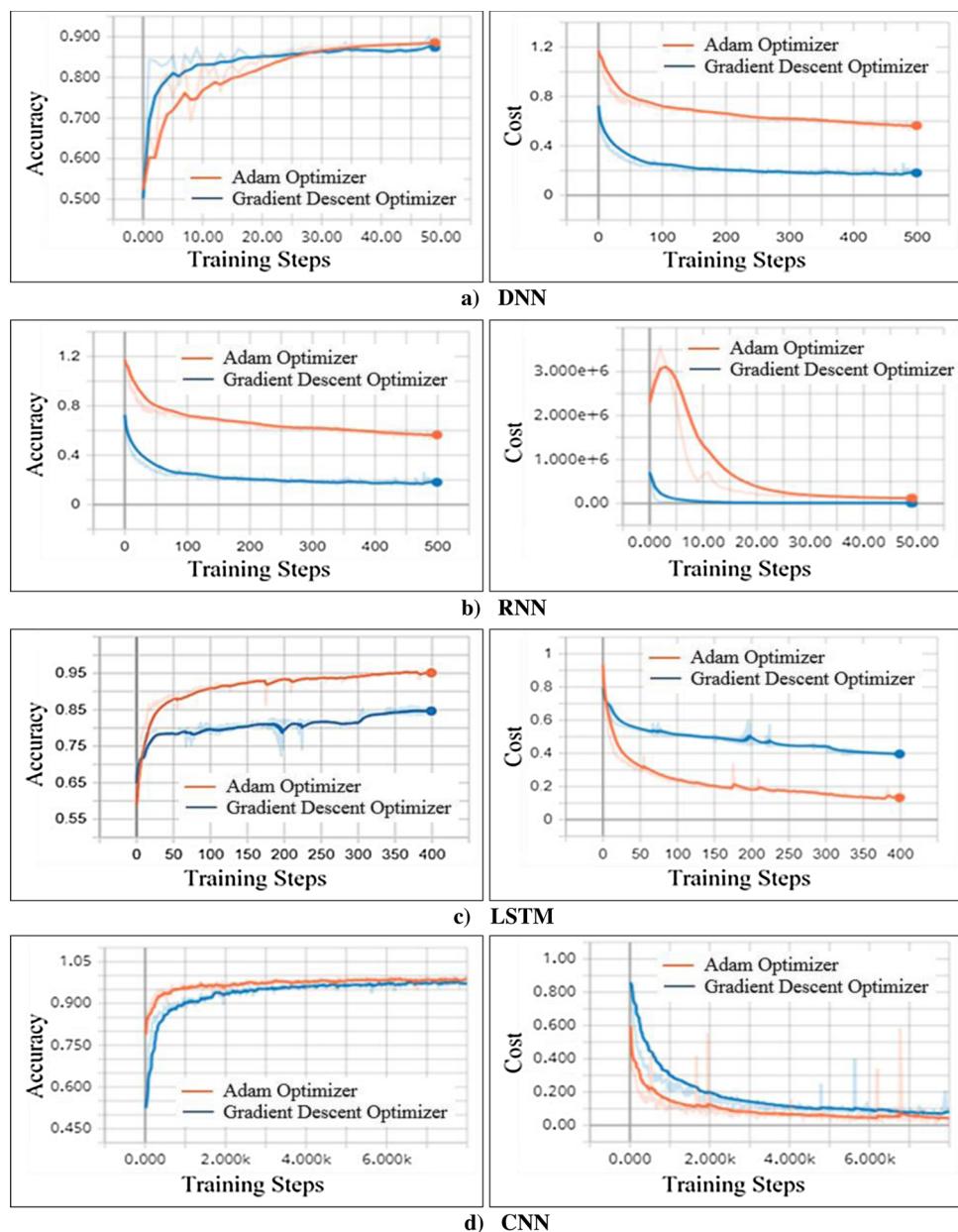


Fig. 8. Training Progress of The Neural Network Models.

**Table 3**

Detection Summary of the Trained Neural Network Models using Gradient Descent Optimizer.

Model	Weather	TP Rate (%)	FN Rate (%)	TN Rate (%)	FP Rate (%)	Overall Accuracy
DNN	Clear	95.0	5.0	90.4	9.6	85.1
	Distant Fog	73.7	26.3	94.4	5.6	
	Near Fog	84.5	15.5	91.1	8.9	
RNN	Clear	90.5	9.5	75.0	25.0	77.4
	Distant Fog	73.2	26.8	81.3	18.7	
	Near Fog	9.2	90.8	98.2	1.8	
LSTM	Clear	84.6	15.4	90.8	9.2	84.2
	Distant Fog	86.7	13.3	82.5	17.5	
	Near Fog	67.9	32.1	98.1	1.9	
CNN	Clear	98.8	1.2	98.9	1.1	97.5
	Distant Fog	98.0	2.0	97.1	2.9	
	Near Fog	86.3	13.8	99.7	0.3	

near fog detection accuracy. In addition, near fog usually occurs over a small area and often creates a dramatic change in visibility. Since both LSTM and RNN utilizes the weights from the previous time steps, any sudden change in time steps might make these models challenging to train. However, the CNN model was not significantly affected by these problems and produced a much-improved near fog detection accuracy of more than 89 %.

Interestingly, the CNN model provided significantly better results compared to the other models, the overall detection accuracy was found to be 98.8 %, 98 %, and 86.3 % for the clear, distant fog, and near fog image groups, respectively. The lowest FN rate was found for the clear image group, where only 1.2 % of the images were misclassified, as can be seen in **Table 3**.

The performance of the neural network models using Adam optimizer is provided in **Table 4**, which shows that the models using this optimizer performed better than the models using Gradient Descent optimizer. The overall detection accuracy of the DNN, RNN, LSTM, and CNN model was found to be 88.4 %, 89.3 %, 92.6 %, 98.1 %,

**Table 4**

Detection Summary of the Trained Neural Network Models using Adam Optimizer.

Model	Weather	TP Rate (%)	FN Rate (%)	TN Rate (%)	FP Rate (%)	Overall Accuracy
DNN	Clear	94.0	6.0	93.2	6.8	88.4
	Distant Fog	86.7	13.3	90.7	9.3	
	Near Fog	61.1	38.9	96.2	3.8	
RNN	Clear	95.8	4.2	94.8	5.2	91.4
	Distant Fog	91.7	8.3	91.2	8.8	
	Near Fog	59.7	4.3	98.6	1.4	
LSTM	Clear	93.5	6.5	97.9	2.1	92.6
	Distant Fog	95.3	4.7	90.5	9.5	
	Near Fog	71.7	28.3	98.9	1.1	
CNN	Clear	99.8	0.2	98.4	1.6	98.1
	Distant Fog	97.6	2.4	98.5	1.5	
	Near Fog	89.1	10.9	99.7	0.3	

respectively. The highest TP rate and the lowest FN rate of the DNN model were found for the clear image group with only 6% misclassification. More specifically, 4% and 2% of the clear images were misclassified as distant fog and near fog images, respectively. Similarly, the TP rate of the distant fog and near fog image group were found to be 86.7 % and 61.1 %, respectively. Considering the RNN model, the highest TP rate was found for the distant fog image group where 95.8 % of the clear fog images were correctly classified. The detection accuracy of the LSTM model using Adam optimizer was found to be 92.6 %, which is higher than the RNN model. The distance fog image group had the highest overall detection accuracy, where more than 95 % of the images were correctly classified. Similarly, the detection accuracy of the clear and near fog image group was found to be 93.5 % and 71.7 %, respectively. In addition, the lowest FP rate of 1.1 % was found for the near fog image group, meaning that 1.1 % of the other images were wrongly classified as near fog images. As expected, the detection accuracy of the trained CNN models was found to be much higher compared to the other neural network models. The trained CNN model provided an outstanding prediction accuracy of 99.8 %, 97.6 %, and 89.1 % for clear, distant fog, and near fog image groups, respectively, as can be seen in Table 4. The false positive (FP) rate of the clear image group using the trained CNN model was found to be only 1.6 %, meaning that 1.6 % of the other images were classified as clear images. It is worth noting that a high FP rate of clear weather is more hazardous since, in such conditions, drivers will be exposed to adverse weather without any warnings. On the other hand, a high FN rate of clear weather will provide frequent adverse weather warnings in clear roadway conditions, which might affect the compliance rate. The FN rate of the trained CNN model for the clear image group was found to be only 0.2 %, meaning only 0.2 % of the clear images were misclassified as other images.

Overall, the prediction accuracy of the CNN models was found to be significantly higher compared to the accuracy of the other NN models. This finding is in line with previous studies. Previous image classification studies using CNN also indicated that CNN is capable of providing far better results than the other types of NN. For instance, Krizhevsky et al. used a subset of the ImageNet database to classify 1000 different classes. ImageNet is a massive image dataset developed for object recognition research and contains over 15 million labeled images belonging to around 22,000 classes. The results of their study showed that CNN is capable of reaching record-breaking accuracy in a highly demanding dataset (Krizhevsky et al., 2012). Several other studies also adopted the CNN structure developed by Krizhevsky et al. with a slight modification to classify the ImageNet dataset and found satisfactory results (Simonyan and Zisserman, 2015; Sermanet et al., 2014).

It is worth mentioning that the test image dataset consisted of images in all traffic conditions, including congested flow. As mentioned earlier, the overall detection accuracy of the CNN model using this test

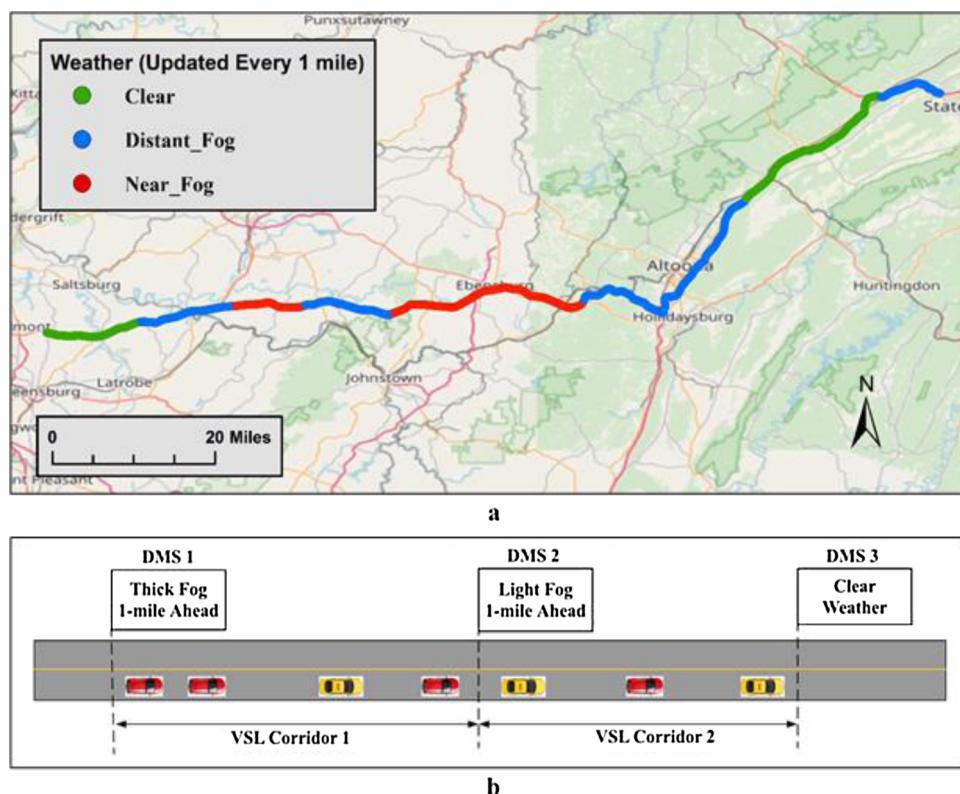
dataset was found to be 98.1 %, which indicates that the proposed fog detection model can detect weather in every traffic condition. In order to further investigate the performance of the model under different traffic conditions, the weather condition of 150 new images in congested traffic (50 in each adverse weather condition) was detected using the developed CNN model. It was found that out of 150 images, the model correctly classified 147 images, indicating that the proposed model can perform equally well in congested traffic.

## 5. Visualization of the proposed model

As discussed in the previous section, the best performing model for detecting fog was found to be CNN with Adam optimizer, which produced an impressive overall accuracy of 98.1 %. In order to visualize the practical application of the proposed model, a NDS trip predominantly traversed on foggy weather on US-22 and I-99 in Pennsylvania were selected and visualized using the ArcGIS software. The length of the trip was around 102 miles, with a total duration of about 1 h and 50 min. The time-series data of the NDS trip was used to get the location (longitude and latitude) of the vehicle, and the video data was used to extract the images of the trips. Twenty images were captured per mile, and using the developed CNN model, the weather was detected for each image. Subsequently, based on the highest vote, the weather condition was determined for that 1-mile segment. Finally, using a color-coded map, the weather for the total trip was visualized, as can be seen in Fig. 9a. Since the proposed weather detection method was based on only a single video camera, it has a huge potential to be a cost-effective way of providing comprehensive weather information for the entire network. In addition, with the evolution of Connected Vehicle technologies, weather data from all the vehicles in a network can be shared with the Traffic Management Centers (TMCs) to create real-time weather-based VSL systems. In addition, with the advancement of tablet/smartphone cameras and the powerful processors they possess, maintenance vehicles could be easily fitted with tablets/smartphones to collect and process geocoded road weather images that could be easily classified via an onboard application.

As mentioned earlier, most of the current weather-based regulatory VSL system is mainly based on the data from weather stations. However, the cost of employing weather stations is very high, and hence, their widespread implementation is not possible. Therefore, for non-VSL corridors, the proposed method has the potential to provide cost-effective advisory VSL. In addition, the method can also be used to disseminate cautionary messages, such as "Thick Fog 1-mile Ahead", within the Advanced Traveler Information System (ATIS) over the DMS to warn the drivers about any potentially hazardous road weather conditions where no weather stations are present. Fig. 9b illustrates a VSL corridor based on geo-specific trajectory-level weather data. To maintain the homogeneity of the speed limit, the weather data from all the available vehicles within a VSL corridor will be considered, and only one representative weather impact index will be used to calculate the speed limits at the beginning of the corridor. For any particular time instance, if the first three vehicles in corridor-1 reported thick fog and the last vehicle in corridor-1 reported light fog, the weather for the whole corridor-1 will be considered as thick fog and will be used to calculate the speed limit at DMS-1. Since the weighted average from all the available vehicle data from corridor-1 will be used, the effect of any false detection will not be that significant. The National Center for Atmospheric Research (NCAR) is also using similar methodologies to reduce random errors during weather forecasting (Delle Monache et al., 2011).

The proposed fog detection method can be implemented using the existing infrastructures and facilities. Currently, many apps, such as Waze, support manual reporting of weather; however, manual reporting could be dangerous because drivers should pay greater attention to the roadway instead of interacting with a mobile app. But the proposed model can automatically detect weather conditions from



**Fig. 9.** Visualization of the Detection Model using Geo-Specific Trajectory Level Weather Data.

video data without any human involvement. The system can be implemented by utilizing the road users' smartphones for automatic detection and reporting of weather conditions. However, regular vehicles might have to cancel their trips under extreme adverse weather conditions, which makes using regular vehicles for weather data collection not always applicable. In such extreme weather conditions, the proposed weather detection model could be a keystone of developing implantable systems for Traffic Management Centers (TMCs) using snowplows and maintenance vehicles as well as integration within the 511 apps for automated weather identification. The maintenance crews from most the Departments of Transportation (DOTs) usually provide service on interstates, high volume highways, principal arterial, and urban routes up to 24 h a day with a goal of maintaining clear roadways for driving safely at reasonable speeds (WYDOT, 2019). In current practice, reporting of real-time road surface weather information is mostly based on snowplow drivers. Snowplow truck drivers manually select a code to describe the prevailing surface weather condition of a road segment based on his/her experience and report the code to the TMCs. Manual reporting, especially in adverse weather, is extremely dangerous because it will create distractions. Therefore, to reduce the risk of manually reporting weather conditions, the maintenance vehicles could be equipped with an automated weather detection system via smartphones/tablets using deep learning and computer vision. The smartphones/tablets will automatically collect and process geocoded road weather images that could be easily classified via an onboard application to get accurate weather conditions, especially in extremely harsh weather when enough regular on-road drivers are not present. Furthermore, the existing roadside webcams or surveillance cameras can also be utilized to implement the weather detection model. Although the proposed detection models are based on the videos captured from moving vehicles, the model can easily be updated using images from existing roadside cameras.

## 6. Conclusions

The primary objective of this study was to develop a method that can detect trajectory-level foggy weather conditions in real-time. The study utilized the massive SHRP2 NDS video data and used a promising machine learning technique known as Neural Network (NN). Python programming on the TensorFlow machine learning library, which is developed by Google, was used for training different models, including Deep Neural Network (DNN), Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM), and Convolutional Neural Network (CNN). First, a database of images was created from the acquired NDS video data. Subsequently, the images were annotated manually and grouped into three categories; clear, distant fog, and near fog; 80 % of which were then used to train the NN models, and the remaining 20 % were used to test the detection accuracy on the models. Two optimizers, including Adam and Gradient Descent, were used during the training process. While the overall prediction accuracy of the DNN, RNN, LSTM, and CNN using the Adam optimizer was found to be 85 %, 77 %, 84 %, and 97 %, respectively, the overall prediction accuracy of the DNN, RNN, LSTM, and CNN considering the Gradient Descent optimizer was found to be around 88 %, 91 %, 93 %, and 98 %, respectively.

The findings from this study can be applied in Connected Variable Speed Limit (VSL) systems, as illustrated previously. The proposed method has the potential to improve roadway safety, especially in adverse weather conditions, including fog. The majority of the current VSL systems are based on data collected from weather stations, which may not provide real-time continuous visibility conditions. In addition, the weather stations are expensive and hence set a limit to their extensive deployment. As the fog detection method developed in this study is capable of providing trajectory-level weather information in real-time with excellent prediction accuracy, in a Connected Vehicle environment, it would improve the VSL systems significantly, especially on roadways with no weather stations.

In addition, with the rapid advancement in connectivity, processing power, and camera quality of the smartphones, the maintenance vehicle

can be equipped with these mobile devices capable of collecting and processing geocoded road weather images that could be easily classified via an onboard application. The road weather images correspond to the images that are taken from inside moving vehicles representing trajectory-level weather conditions at road surface level from driver perspective. From the roadway safety point of view, weather conditions at the road surface levels are more important than the atmospheric weather. It is worth mentioning that the proposed fog detection system closely resembles mobile mapping technology. Mobile mapping is the method of compiling geospatial data from a mobile vehicle, typically equipped with an array of different sensors such as camera, radar, laser, and LiDAR (Escalera and Radeva, 2004). However, in this study, we are proposing the use of smartphone cameras from inside a vehicle as a source of moving weather station.

Results from this study can also be used to develop an affordable Advanced Driving Assistance System (ADAS). A simple mobile phone application with proper data connectivity can effectively be fitted in maintenance vehicles and used to detect roadway fog and visibility conditions in real-time, and real-time data could be disseminated to Traffic Management Centers (TMCs). The Machine Vision methodology provided in this research could be extended to identify work zones, pedestrians, crashes, road closures, and so forth. With the evolution of the Connected Vehicle technologies and 5G communications, these data can easily be shared with other vehicles in a Vehicle-to-Vehicle (V2V) environment for the provision of appropriate warnings and related information based on real-time conditions of the roadway. Therefore, the system proposed in this study has the potential to provide affordable ADAS with multiple functions on a single hardware platform without using costly sensors.

It is worth mentioning that, despite having groundbreaking achievements in many fields, including artificial intelligence, deep learning has several limitations. Deep learning models may uncover the latent patterns in the image pixels, but they cannot understand what the patterns actually mean. Therefore, deep learning can provide excellent performance for properly annotated benchmark datasets but sometimes can perform poorly for slightly different real-world data. To overcome this problem, deep learning, especially supervised learning, requires a large amount of annotated data in every possible scenario, which is extremely hard to get considering the fact that annotated images are not readily available and human-curation and/or human annotation of a training data set is a challenging and time-consuming task. Therefore, in order to reduce the need for supervision, future studies will explore the potential of using other variations of deep learning methods, such as unsupervised learning, weakly supervised learning, semi-supervised learning, and transfer learning (Goodfellow et al., 2016). Another limitation of deep neural networks is weight adjustment during training. The weights are adjusted in each iteration using optimizers; however, there is no rule of thumb what optimizer will provide the best possible set of weights for a particular dataset. Although this study utilized two optimizers (i.e., Gradient descent and Adam), future research will also include other optimizers to improve the performances of the deep neural networks by finding the best set of weights.

## Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## CRediT authorship contribution statement

**Md Nasim Khan:** Conceptualization, Methodology, Software, Formal analysis, Validation, Writing - original draft, Writing - review & editing. **Mohamed M. Ahmed:** Conceptualization, Methodology, Software, Validation, Writing - review & editing, Supervision, Project administration, Funding acquisition.

## Acknowledgments

This work was conducted under the second Strategic Highway Research Program (SHRP2), which is administrated by the Transportation Research Board of the National Academies, and it was sponsored by the Federal Highway Administration in cooperation with the American Association of State Highway and Transportation Officials (AASHTO), and the Wyoming Department of Transportation (WYDOT).

## References

- Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., Kudlur, M., Levenberg, J., Monga, R., Moore, S., Murray, D.G., Steiner, B., Tucker, P., Vasudevan, V., Warden, P., Wicke, M., Yu, Y., Zheng, X., 2016. TensorFlow: a system for large-scale machine learning. *OSDI* 16, 265–283.
- Abdel-Aty, M., Ekram, A.A., Huang, H., Choi, K., 2011. A study on crashes related to visibility obstruction due to fog and smoke. *Accid. Anal. Prev.* 43 (5), 1730–1737. <https://doi.org/10.1016/j.aap.2011.04.003>.
- Ahmed, M.M., Abdel-Aty, M., Lee, J., Yu, R., 2014. Real-time assessment of fog-related crashes using airport weather data: a feasibility analysis. *Accid. Anal. Prev.* 72, 309–317. <https://doi.org/10.1016/j.aap.2014.07.004>.
- Ahmed, M., Young, R., Ghasemzadeh, A., Hammit, B., Ali, E., Khan, N., Das, A., Eldeeb, H., 2017. Implementation of SHRP2 results within the Wyoming connected vehicle variable speed limit system: Phase 2 early findings report and phase 3 proposal. WYDOT.
- Ahmed, M.M., Ghasemzadeh, A., Hammit, B., Khan, M.N., Das, A., Ali, E., Young, R.K., Eldeeb, H., 2018. Driver Performance and Behavior in Adverse Weather Conditions: An Investigation Using the SHRP2 Naturalistic Driving Study Data-Phase 2. Final Report WY-18/05F.
- Alotaibi, M., Alotaibi, B., 2019. Distracted driver classification using deep learning. *Signal Image Video Process.* <https://doi.org/10.1007/s11760-019-01589-z>.
- Ali, E.M., Ahmed, M.M., Wulff, S.S., 2019. Detection of critical safety events on freeways in clear and rainy weather using SHRP2 naturalistic driving data: Parametric and non-parametric techniques. *Saf. Sci.* 119, 141–149. <https://doi.org/10.1016/j.ssci.2019.01.007>.
- Bao, J., Liu, P., Ukkusuri, S.V., 2019. A spatiotemporal deep learning approach for city-wide short-term crash risk prediction with multi-source data. *Accid. Anal. Prev.* 122 (October), 239–254. <https://doi.org/10.1016/j.aap.2018.10.015>.
- Bronte, S., Bergasa, L.M., Alcantarilla, P.F., 2009. Fog detection system based on computer vision techniques. In: 2009 12th International IEEE Conference on Intelligent Transportation Systems. November. <https://doi.org/10.1109/ITSC.2009.5309842>.
- Colding, P.J., 1971. Thick Fog and Its Effect on Traffic Flow and Accident. *Transport and Road Research Laboratory TRRL Report LR 397*.
- Das, A., Ahmed, M.M., 2019. Exploring the effect of fog on lane-changing characteristics utilizing the SHRP2 naturalistic driving study data. *J. Transp. Saf. Secur.* <https://doi.org/10.1080/19439962.2019.1645777>.
- Das, A., Ghasemzadeh, A., Ahmed, M.M., 2019. Analyzing the effect of fog weather conditions on driver lane-keeping performance using the SHRP2 naturalistic driving study data. *J. Safety Res.* 68, 71–80. <https://doi.org/10.1016/j.sjr.2018.12.015>.
- Das, A., Khan, M.N., Ahmed, M.M., 2020a. Nonparametric multivariate adaptive regression splines models for investigating lane-changing gap acceptance behavior utilizing strategic highway research program 2 naturalistic driving data. *Transp. Res. Rec.* 1–16. <https://doi.org/10.1177/0361198120914293>.
- Das, A., Khan, M.N., Ahmed, M.M., 2020b. Detecting lane change maneuvers using SHRP2 naturalistic driving data: A comparative study machine learning techniques. *Accid. Anal. Prev.* 142.
- Delle Monache, L., Nipen, T., Liu, Y., Roux, G., Stull, R., 2011. Kalman Filter and Analog Schemes to Postprocess Numerical Weather Predictions. pp. 3554–3570.
- Donges, Niklas. Gradient Descent in a Nutshell – Towards Data Science. <https://towardsdatascience.com/gradient-descent-in-a-nutshell-eaf8c18212f0>. (Accessed 25 July 2018).
- Duchi, J., 2011. Adaptive subgradient methods for online learning and stochastic optimization. *J. Mach. Learn. Res.* 12, 2121–2159.
- Escalera, S., Radeva, P., 2004. Fast greyscale road sign model matching and recognition. *Recent Adv. Artif. Intell. Res. Dev.*, vol. 2 (1), 69–76.
- FHWA. How Do Weather Events Impact Roads? - FHWA Road Weather Management. [https://ops.fhwa.dot.gov/weather/q1\\_roadimpact.htm](https://ops.fhwa.dot.gov/weather/q1_roadimpact.htm). (Accessed 20 September 2019).
- Formosa, N., Quddus, M., Ison, S., Abdel-Aty, M., Yuan, J., 2020. Predicting real-time traffic conflicts using deep learning. *Accid. Anal. Prev.* 136 (July). <https://doi.org/10.1016/j.aap.2019.105429>.
- Goodfellow, I., Bengio, Y., Courville, A., 2016. Deep Learning. MIT Press.
- Google. TensorFlow. <https://opensource.google.com/projects/tensorflow>. (Accessed 26 July 2018).
- Google. TensorBoard: TensorFlow's Visualization Toolkit.
- Graves, Alex, Abdel-rahman Mohamed, G.H., 2013. Speech Recognition with Deep Recurrent Neural Networks. [arXiv:1303.5778](https://arxiv.org/abs/1303.5778).
- Hankey, J.M., Perez, M.A., McClafferty, J.A., 2016. Description of the SHRP 2 naturalistic database and the crash, near-crash, and baseline data sets. Virginia Tech Transportation Institute.
- Hasemi, M., Mirashid, A., Shirazi, A.B., 2020. Deep Learning-Based Driver Distraction and Drowsiness Detection. [arXiv preprint arXiv:2001.05137](https://arxiv.org/abs/2001.05137).

- Hassan, H.M., Abdel-aty, M.A., 2013. Predicting reduced visibility related crashes on freeways using real-time traffic flow data. *J. Safety Res.* 45, 29–36. <https://doi.org/10.1016/j.jsr.2012.12.004>.
- Hautière, N., Labayrade, R., Aubert, D., 2006a. Real-time disparity contrast combination for onboard estimation of the visibility distance. *IEEE Trans. Intell. Transp. Syst.* 7 (2), 201–211. <https://doi.org/10.1109/TITS.2006.874682>.
- Hautière, N., Tarel, J.P., Lavenant, J., Aubert, D., 2006b. Automatic fog detection and estimation of visibility distance through use of an onboard camera. *Mach. Vis. Appl.* 17 (1), 8–20. <https://doi.org/10.1007/s00138-005-0011-1>.
- Huang, T., Wang, S., Sharma, A., 2020. Highway crash detection and risk estimation using deep learning. *Accid. Anal. Prev.* 135 (April), 105392. <https://doi.org/10.1016/j.aap.2019.105392>.
- Ibrahim, M.R., Haworth, J., 2019. WeatherNet: recognising weather and visual conditions from street-level images using deep residual learning. *Int. J. Appl. Earth Obs. Geoinf.* 8 (12), 549.
- Jacobé de Naurois, C., Bourdin, C., Stratulat, A., Diaz, E., Vercher, J.L., 2019. Detection and prediction of driver drowsiness using artificial neural network models. *Accid. Anal. Prev.* 126 (November), 95–104. <https://doi.org/10.1016/j.aap.2017.11.038>.
- Jeon, H., Lee, J., Sohn, K., 2018. Artificial intelligence for traffic signal control based solely on video images. *J. Intell. Transp. Syst.* 0 (0), 1–13. <https://doi.org/10.1080/15472450.2017.1394192>.
- Karn, U. Introduction to Neural Networks. <https://ujjwalkarn.me/2016/08/09/quick-intro-neural-networks/>. (Accessed 24 July 2018).
- Khan, M.N., Ahmed, M.M., 2019. Snow detection using in-vehicle video camera with texture-based image features utilizing K-Nearest neighbor, support vector machine, and random forest. *Transp. Res. Rec.* 2673 (8), 221–232. <https://doi.org/10.1177/0361198119842105>.
- Khan, M.N., Ghasemzadeh, A., Ahmed, M.M., 2018. Investigating the impact of fog on freeway speed selection using the SHRP2 naturalistic driving study data. *Transp. Res. Rec.* 2672 (16), 93–104. <https://doi.org/10.1177/0361198118774748>.
- Kingma, D.P., Lei, J., Adam, B., 2014. A Method for Stochastic Optimization. [arXiv:1412.6980](https://arxiv.org/abs/1412.6980), .
- Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. ImageNet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* 1097–1105.
- Krummenacher, G., Ong, C.S., Koller, S., Kobayashi, S., Buhmann, J.M., Member, S., 2018. Wheel defect detection with machine learning. *IEEE Trans. Intell. Transp. Syst.* 19 (4), 1176–1187. <https://doi.org/10.1109/TITS.2017.2720721>.
- Li, L., Qu, X., Zhang, J., Wang, Y., Ran, B., 2019. Traffic speed prediction for intelligent transportation system based on a deep feature fusion model. *J. Intell. Transp. Syst.* 24(50). <https://doi.org/10.1080/15472450.2019.1583965>.
- Li, P., Abdel-Aty, M., Yuan, J., 2020. Real-time crash risk prediction on arterials based on LSTM-CNN. *Accid. Anal. Prev.* 135.
- Lipton, Z.C., Berkowitz, J., Elkan, C., 2015. A Critical Review of Recurrent Neural Networks for Sequence Learning. pp. 1–38. <https://doi.org/10.1145/2647868.2654889>.
- Lv, Y., Duan, Y., Kang, W., Li, Z., Wang, F., 2015. Traffic flow prediction with big data: a deep learning approach. *IEEE Trans. Intell. Transp. Syst.* 16 (2), 865–873. <https://doi.org/10.1109/TITS.2014.2345663>.
- Mueller, F.H., 1959. History of Observational Instructions on Fog. *Key Meteorological Records Documentation No. 3.031*. pp. 1–8.
- Murphy, R., Swick, R., Hamilton, B.A., Guevara, G., 2012. Best Practices for Road Weather Management. Version 3.0.
- Napoletano, P., Schettini, R., 2018. Recognition of driver distractions using deep learning. *2018 IEEE 8th International Conference on Consumer Electronics - Berlin (ICCE-Berlin)* 1–6.
- Pavlić, M., Belzner, H., Rigoll, G., Ilić, S., 2012. Image based fog detection in vehicles. *IEEE Intell. Veh. Symp.* 1132–1137. <https://doi.org/10.1109/IVS.2012.6232256>.
- Pomerleau, D., 1997. Visibility estimation from a moving vehicle using the RALPH vision system. *Proc. IEEE Conf. Intell. Transp. Syst.* doi:10.1109/ITSC.1997.660594.
- Prabhat, M., Racah, E., Biard, J., Liu, Y., Mudigonda, M., Kashinath, K., Beckham, C., Maharaj, T., Kahou, S., Pal, C., O'Brien, T.A., Wehner, M.F., Kunkel, K., Collins, W.D., 2017. Deep Learning for Extreme Weather Detection. *Am. Geophys. Union Fall Meet.*
- Ruder, S., 2016. An Overview of Gradient Descent Optimization. [arXiv:1609.04747](https://arxiv.org/abs/1609.04747), pp. 1–14.
- Schechner, Y.Y., Narasimhan, S.G., Nayar, S.K., 2001. Instant dehazing of images using polarization. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. *IEEE Comput. Soc.* pp. 325–332. <https://doi.org/10.1109/CVPR.2001.990493>.
- Sebastiani, F., 2002. Machine learning in automated text categorization. *ACM Comput. Surv. (CSUR)* 34 (1), 1–47.
- Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., Lecun, Y., 2014. OverFeat: integrated recognition, localization and detection using convolutional networks. *ICLR*.
- Shwartz, S., Namer, E., Schechner, Y.Y., 2006. Blind haze separation. *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Vol. 2 1984–1991. <https://doi.org/10.1109/CVPR.2006.71>.
- Simonyan, K., Zisserman, A., 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv preprint arXiv:1409.1556*, .
- Smadi, O., Hawkins, N., Hans, Z., Bektas, B.A., Knickerbocker, S., Nlenanya, I., Souleyrette, R., Hallmark, S., 2015. Naturalistic Driving Study: Development of the Roadway Information Database. *SHRP 2 Report: S2-S04A-RW-1*, p. 108.
- Stone, R.G., 1936. Fog in the United States and adjacent regions. *Geogr. Rev.* 26 (1), 111–134.
- Tan, R.T., 2008. Visibility in bad weather. *Computer Vision and Pattern Recognition CVPR 2008*. pp. 1–8.
- Tang, K., Chen, S., Khattak, A.J., Pan, Y., 2019. Deep architecture for citywide travel time estimation incorporating contextual information deep architecture for citywide travel time estimation incorporating. *J. Intell. Transp. Syst.* 0 (0), 1–17. <https://doi.org/10.1080/15472450.2019.1617141>.
- TensorFlow. Cross-Entropy with Logit. [https://www.tensorflow.org/api\\_docs/python/tf/nn/sigmoid\\_cross\\_entropy\\_with\\_logits](https://www.tensorflow.org/api_docs/python/tf/nn/sigmoid_cross_entropy_with_logits). (Accessed 25 July 2018).
- Tieleman, T., Hinton, G., 2012. Lecture 6.5-Rmsprop, Coursera: Neural Networks for Machine Learning. Technical Report.
- Usman, T., Fu, L., Miranda-moreno, L.F., 2010. Quantifying safety benefit of winter road maintenance: accident frequency modeling. *Accid. Anal. Prev.* 42 (6), 1878–1887. <https://doi.org/10.1016/j.aap.2010.05.008>.
- Wang, J., Kong, Y., Fu, T., 2019. Expressway crash risk prediction using back propagation neural network: a brief investigation on safety resilience. *Accid. Anal. Prev.* 124 (December), 180–192. <https://doi.org/10.1016/j.aap.2019.01.007>.
- Wu, Y., Abdel-Aty, M., Lee, J., 2018. Crash risk analysis during fog conditions using real-time traffic data. *Accid. Anal. Prev.* 114 (May), 4–11. <https://doi.org/10.1016/j.aap.2017.05.004>.
- WYDOT, 2019. Wyoming Report on Traffic Crashes – 2018. *Highway Safety Program, Wyoming Department of Transportation, Cheyenne, WY*.
- Xu, M., Wu, J., Huang, L., Zhou, R., Wang, T., Hu, D., 2020. Network-wide traffic signal control based on the discovery of critical nodes and deep reinforcement learning. *J. Intell. Transp. Syst. Technol. Plan. Oper.* 24 (1), 1–10. <https://doi.org/10.1080/15472450.2018.1527694>.
- Yu, R., Abdel-aty, M., Ahmed, M., 2013. Bayesian random effect models incorporating real-time weather and traffic data to investigate mountainous freeway hazardous factors. *Accid. Anal. Prev.* 50, 371–376. <https://doi.org/10.1016/j.aap.2012.05.011>.
- Yu, R., Xiong, Y., Abdel-aty, M., 2015. A correlated random parameter approach to investigate the effects of weather conditions on crash risk for a Mountainous Freeway. *Transp. Res. C 50*, 68–77. <https://doi.org/10.1016/j.trc.2014.09.016>.
- Zhang, D., Sullivan, T., O'Connor, N.E., Gillespie, R., Regan, F., 2015. Coastal fog detection using visual sensing. *OCEANS 2015-Genova*, IEEE, Vol. 1–5.