CETRA^{2020*} 20-21 May 2021, Zagreb, Croatia
6th International Conferenceon Road and Rail Infrastructure

UNCERTAINTY ESTIMATION ON ROAD SAFETY ANALYSIS USING BAYESIAN DEEP NEURAL NETWORKS

Pan Guangyuan[1,2], Chen Qili[3], Fu Liping[4,5], Yu Ming[2], Muresan Matthew[4]
[1] *Shenzhen Garry Intelligent Technology ltd, Shenzhen, China*
[2] *The Chinese University of Hong Kong, Department of Electronic Engineering, Hong Kong SAR*
[3] *Beijing University of Technology, Faculty of Information Technology, Beijing, China*
[4] *University of Waterloo, Department of Civil & Environmental Engineering, Canada*
[5] *Wuhan University of Technology, Intelligent Transportation Systems Research Centre, China*

Abstract

Deep neural networks have been successfully used in many different areas of traffic engineering, such as crash prediction, intelligent signal optimization and real-time road surface condition monitoring. The benefits of deep neural networks are often uniquely suited to solve certain problems and can offer improvements in performance when compared to traditional methods. In collision prediction, uncertainty estimation is a critical area that can benefit from their application, and accurate information on the reliability of a model's predictions can increase public confidence in those models. Applications of deep neural networks to this problem that consider these effects have not been studied previously. This paper develops a Bayesian deep neural network for crash prediction and examines the reliability of the model based on three key methods: layer-wise greedy unsupervised learning, Bayesian regularization and adapted marginalization. An uncertainty equation for the model is also proposed for this domain for the first time. To test the performance, eight years of car collision data collected from Highway 401, Canada, is used, and three experiments are designed.

Keywords: road safety, road surface condition, crash prediction, neural networks

1 Introduction

The technology of deep neural networks (DNNs) is becoming ubiquitous in both research and industry fields. By stacking layers of linear convolutions with appropriate non-linearities, abstract concepts can be learnt from high-dimensional input alleviating the challenging and time-consuming task of hand-crafting algorithms [1-2]. With this advance, almost in each passing month, new applications on a variety of problems like unsupervised-learning based classification, object detection, forecasting, and reinforcement learning based game playing, continue to be found by using DNNs [3-6]. Moreover, in recent years, a number of deep learning-based methods have also been applied in traffic engineering, such as road surface condition monitoring, traffic sign recognition, autopilot, intelligent traffic optimization, etc [7-10]. In the domain of road safety analysis, deep learning-based models have been shown as effective when compared to the traditional negative binomial and to other techniques such as kernel regression [12]. In this paper, we focus on solving the problem of predicting collisions for highways and achieving high accuracy and fast training speed. We have also explored the relation between the model size, data size and trained model accuracy. While previous researches have done similar works using traditional artificial neural networks

DOI: https://doi.org/10.5592/CO/CETRA.2020.1218

(ANNs), surprisingly, DNN-based solutions for road safety applications have so far been suggested without any uncertainty management [10-12]. This study aims to fill this gap in the literature by quantifying and estimating the uncertainty of DNN predictions.

Information about the reliability of automated predictions is a key requirement for them to be integrated into alarm systems for the public. No matter whether data is short or abundant, difficult and unseen scenarios are unavoidable. Therefore, DNNs should report, in addition to the decision, an associated estimate of uncertainty. Estimating the uncertainty from a DNN-based prediction on a single testing sample requires a distribution over possible outcomes, for which a Bayesian perspective is best suited. Indeed, Bayesian approaches to uncertainty estimation have been proposed to assess the reliability of model predictions in many domains [13-15]. The integration of the Bayesian methods and DNNs is an active topic of research, but the practical value of the developed methods has yet to be demonstrated in traffic engineering. There are two major types of uncertainty, aleatoric uncertainty and epistemic uncertainty. The first one captures noise inherent in the observations, while the later one accounts for uncertainty in the model which can be explained away given enough data. Aleatoric uncertainty is unavoidable, while epistemic uncertainty is more practical and has been mostly focused on [16-18]. Our previous attempts have focussed on deep learning models and their applications. Specifically, in the domain of road safety analysis, a global DNN model was introduced as an alternative to the traditional regression models for crash modelling [12]. An extensive empirical study was conducted using three real world crash data sets covering six classes of highways as defined by location (urban vs. rural), number of lanes, access control, and region. As an improvement in previous work in this area, this paper proposes a Bayesian deep neural network (BDNN) version that analyses the stability and epistemic uncertainty automatically and gives more reasonable suggestions.

Our focus in this paper is to propose a Bayesian regularization based deep neural network for road safety analysis. The core techniques of this model are, 1) layer-wise greedy unsupervised learning used for data feature learning; 2) Bayesian regularization for knowledge distribution management and uncertainty estimation; and 3) the use of a specifically designed output layer called marginalization which uses the Softmax function predict the probabilities and uncertainties for different scenarios. This paper's structure is organized as follows: Section 2 introduces our methodology, including the improvements on the model and the equation to quantify model's uncertainty; Section 3 demonstrates a case study based on our previous research, more details on implementing the model will be explained; Section 4 summarises the study as well as directions in the future.

2 Methodology

2.1 The training of Bayesian Deep Neural Networks

The structure of proposed model is shown in Figure 1. In the model, three parts are included. The first layer is input layer, it receives sampling values from the original road observed features. Two hidden layers are placed after the input layer and are used for feature extraction. Finally, the output from the hidden layers is marginalised and fed to the output layer. The marginalization layer helps to resort the prediction from exact collision amounts to risk-level based distributions, and uses Softmax to output the result as a percentage for each level.

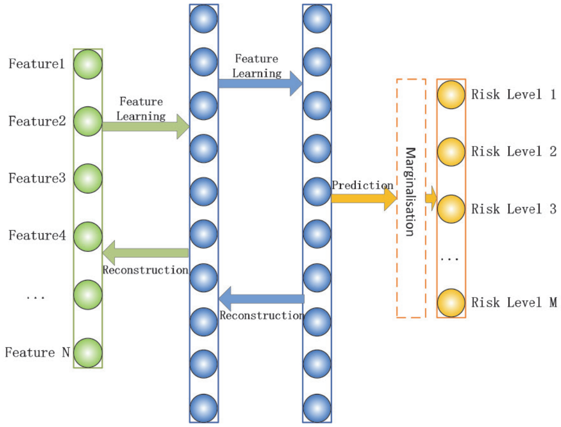

Figure 1 The structure of Bayesian Deep Neural Networks

The training of a BDNN has two steps, the first step is a layer by layer greedy unsupervised learning in which each two neighbouring layers form a restricted Boltzmann machine (RBM). Each RBM has a two-way structure with full connectivity and no specific connections between each same layer. When training each RBM, the hidden layer (the later layer of an RBM) extracts features and information from input to form an order and more abstract representation. Features are transferred in this way until they pass through the last layer and reach the output. The following equations are form the basis of how input is transformed in RBMs and how training takes place:

$$y_j = \varphi_j\left(\sum_j w_{ij}x_i + \sigma \cdot N_j(0,1)\right) \tag{1}$$

$$x_i = \varphi_i\left(\sum_j w_{ij}y_j + \sigma \cdot N_i(0,1)\right) \tag{2}$$

$$\varphi(X) = \theta_L + (\theta_H - \theta_L)\cdot\frac{1}{1+e^{-a_i X}} \tag{3}$$

x_i and y_i is the continuous value of unit i in input layer and j in output layer; w_{ij} is the weight between them, N(0,1) is a Gaussian random variable with mean 0 and variance 1; σ is a constant; ϕ(X) denotes a sigmoid-like function with asymptote of θ_H and θ_L; and a is a variable that controls noise, which means it controls the gradient of the transfer function. When training an RBM, the states are recorded as X^0 (input values, the vector of x_i), Y^0 (hidden layer state using equation 1, the vector of y_j), X^1 (input layer state using equation 2), and Y^1(hidden layer state using equation 1 again). The weights' updating function is

$$W^{(t+1)} = W^{(t)} - \left(X^0 \cdot Y^0 - X^1 \cdot Y^1\right) \tag{4}$$

In the second step, the BDNN is fine-tuned in terms of model structure and learning rate. In doing this, the network will be unfolded, while keeping the trained weights unchanged, to a multilayer back propagation network that uses a gradient descent algorithm to fine-tune the weights. Bayesian regularization is implemented using the following equations:

$$F_W = \alpha E_W + \beta R_W \tag{5}$$

$$E_W = \frac{1}{T}\sum_{t=1}^{T}\left(Y_{target} - Y_{observed}\right)^2 \tag{6}$$

$$R_W = \frac{1}{I \cdot J}\sum_{j=1}^{J}\sum_{i=1}^{I} w_{ij}^2 \tag{7}$$

E_W is the traditional objective function used in a back-propagation network, which calculates the error between target output Y_{target} and observed output $Y_{observed}$ T the testing set; R_W is the Bayesian regularization item that controls the weight between two layers to prevent it from becoming too big in iterations; and α and β are called performance parameters that can be calculated during the iteration. If α›β , then the first part of F_W dominates, which means that the objective of the training is to decrease the training error, if α‹β , the training will focus on decreasing the weights. Finally, I and J are the amount of units in each layer. The weights updating algorithm is,

$$W_k^{t+1} = W_k^t - \Delta W \tag{8}$$

$$\Delta W = \frac{\partial F_W}{\partial W_k} = \alpha \frac{\partial E_W}{\partial W_k} + \beta \frac{\partial R_W}{\partial W_k} \tag{9}$$

$$\frac{\partial E_W}{\partial W_k} = \frac{\partial F_W}{\partial Y_n} \cdot \frac{\partial Y_n}{\partial X_{n-1}} \cdot \frac{\partial X_{n-1}}{\partial Y_{n-2}} \cdots \frac{\partial X_k}{\partial W_k} \tag{10}$$

$$\frac{\partial R_W}{\partial W_k} = \frac{2}{I \cdot J}\sum_{j=1}^{J}\sum_{i=1}^{I} w_{ij} \tag{11}$$

n is the number of layers, so Y_n is the output of the whole network, $Y_n = Y_{observed}$; X_k is the vector of layer k. Therefore, by introducing this regularization term, we can expect that in an iteration, if one weight is affected by its neighbour, which should not be happening, the change of amount of the affection will be reduced using equations (8-11). Furthermore, using this method will ensure weights that do not contribute to the response will be minimized, keeping the network in a sparse connection state.

2.2 Marginalization and Uncertainty Estimation Equations

Marginalization layer is another technique adopted in BDNN to ensure realistic output. To perform this, in training and testing, if the prediction in a category is 0 % while the neighbouring ones are not, the predictions on each category will be re-distributed using equation (12). After training, the marginalization index in equation (13) is applied to estimate how well the distributions are. Finally, when testing the model, testing accuracy can be estimated using equation (14), and the uncertainty with equation (15).

$$P\left(y_i\right) = \partial_1 \cdot P\left(y_{i-1}\right) + \partial_2 \cdot PP\left(y_{i+1}\right) \tag{12}$$

$$MI = f(P\left(y_i\right) > \delta) \tag{13}$$

$$Acc = \frac{1}{M}\sum\left(y_i - \hat{y}_i\right)^2 \tag{14}$$

$$Mu = \frac{1}{M} \cdot \frac{MI - T \cdot Acc}{R \cdot T} \tag{15}$$

$P(y_i)$ is the probability of car collision on risk level i; ∂_1 and ∂_2 are the percentage probabilities taken from the specific categories; $f(P(y_i) > \delta)$ means the total case amount that $P(y_i)$ is over δ; T is the total number of observations and R is the total output neurons; MI is the marginalization index; M is the running times of the model; y_i the ground truth; $\hat{y}_i$ is the integrated confidence; Mu is the marginalized model uncertainty and Acc is the marginalized model accuracy.

Traditionally, Acc (in some papers MAE or RMSE are used) is the only standard to evaluate a model and the higher Acc the better. However, this can obscure the true accuracy of a model in some circumstances. For example, in BDNN, the output is the probabilities for each risk level (explained in next section) and thus if Acc is the only evaluation used then the model will try to even the prediction on each category, which means that during the training the model will try to make a tie between all the categories, if so the truth ground will definitely fall into a category and it will be 100 % correct. This prediction, however, is totally meaningless because it cannot provide any alarms or suggestions to the public and department for management. Mu is a compliment to Acc because it shows how much confidence the model has over a certain observation and how stable the model's prediction is. The lower Mu the better because a lower Mu means the distribution on the prediction is more confident. Thus, in this research, we propose to utilise both Acc and Mu for model evaluation.

3 Experiment

3.1 Experimental Design

This case study is based on historical collisions and related data from Highway 401 in Ontario, Canada. This highway, which plays a crucial role for the region's socio-economic development, is one of the busiest highways in North America. The total length of the highway is 817.9 km of which approximately 800 km is used in this study. According to 2008's traffic volume data, the annual average daily traffic (AADT) ranges from 14,500 to 442,900 indicating comparatively a very busy road corridor. Traffic count data consists of AADT and average annual commercial vehicle counts for the period 2000 - 2008. As each observation records the LHRS and offset information, traffic counts can be spatially located using a linear referencing GIS tool. The whole highway is divided into 418 homogeneous sections (HS) with length ranging from 0.2 km to 12.7 km. Each HS is then assigned the nearest traffic observation. Finally, the processed crash and traffic data are integrated into a single dataset with HS and year as the mapping fields, resulting in a total of 3762 records. The selected input features used in the study are exposure, AADT, left of shield, median width, right of shield, and curve deflection.

In this case study, the model's performance is estimated based on accuracy (Acc), model uncertainty (Mu) and marginalization index (MI), as defined in previous equations. Three experiments are designed below, to testify the performance of the model. We use a network structure of 6-10-10-4 (6 units for the input layer - 10 units for the hidden layer 1 - 10 units for the hidden layer and 2 - 4 units for the output layer) for the model. This structure is selected on the basis of our previous work, which finds that when the training dataset is around 3,000 two hidden layers with no more than 10 are the most effective. Also, different learning rates varying from 0.5 to 3 are tried, with the learning rate of 1.0 being the most effective. In terms of training, we use a total of 50 unsupervised iterations and 1000 fine-tuning iterations to ensure model convergence. Six input neurons are used based on six key features: exposure, AADT, left of shield, median width, right of shield, and curve deflection. Four output neurons are provided corresponding to four risk levels designed based on the annual collision rate. Risk level 1 covers sections with less than 3 collisions per year, risk level 2 is covers sections with 4 to 10 collisions per year, risk level 3 covers sections with 11 to 20 collisions per year and risk level 4 covers sections with over 21 collisions each year. The data distribution is shown in figure 2. In calculating MI, we set $\partial_1 = \partial_2 = 2$.

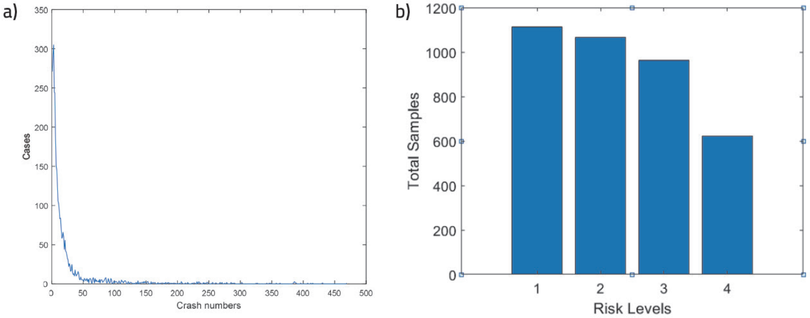

Figure 2 a) crash amount on each segmentation observed each year, b) the re-sorted risk levels based on (a)

3.2 Uncertainty Test with Pre-set Marginalization

A total eight years of collision data with 3762 sets are collected. The first 6 years with 2926 sets are used for model training and the rest 836 sets are for model testing. The marginalization parameters (∂_1, ∂_2) are set to be 2. Other model parameters have been given in last section. The results are shown in Table 1, Table 2 and Figure 4.

Table 1 shows the predictions on ten selected samples. When comparing the predicting risk with the exact risk level (or collision number) using traditional models, BDNN outputs the probabilities on each level and evens the distributions more realistically, thanks to marginalization and Softmax. For some cases (2 and 3), BDNN outputs are given with a very high confidence level, while for most other samples, it is more conservative and provides more possibilities for the risk level. This characteristic allows the model to provide a complete picture that is clearer, such as the situation on cases 5 and 6, providing better results than traditional models that often fail to capture these situations. In table 2, the advances of using BDNN is clearer. In our previous work, the R-DBN-based proposed deep learning model had the best performance when compared to other popular methods. However, when predicting risk levels, the model only achieves an accuracy level of 61 % because the uncertainty is too high. It is outperformed significantly by BDNN, which has a higher accuracy of 83.35 % and an uncertainty value (Mu) only 0.2344, which is a clear sign of better stability. Full comparison results are shown in Table 2, along with results for BDNN models with different confidence ranges, and without marginalisation. The finding proves that the proposed model receives lower prediction accuracy, but higher stability if it is restricted, which is quite reasonable. Figure 3 shows the results of table 1 and 2. The blue lines in both sub-figures are the ground truth. Red line in (a) represents the exact predictions using R-DBN, and the green band in (b) shows the confidence area.

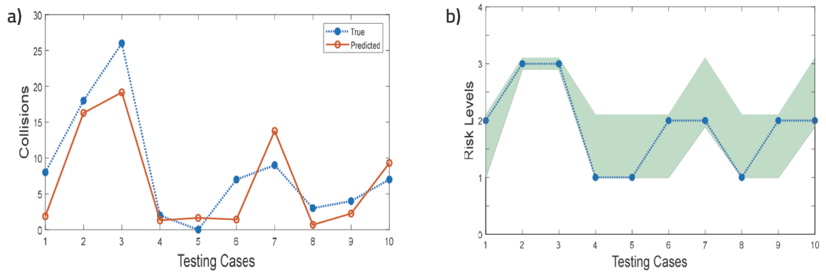

Figure 3 Comparisons on randomly selected testing samples.

Table 1 Predictions on Randomly Selected Testing Samples

Testing Cases		**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Ground truth on risk levels		**2**	**3**	**3**	**1**	**1**	**2**	**2**	**1**	**2**	**2**
Predicted risk levels (%)	1	13.5	0.0	0.0	99.6	9.4	90.6	0.0	67.4	45.0	0.0
	2	86.5	0.0	0.0	0.4	90.6	9.4	99.8	32.6	55.0	99.3
	3	0.0	100.0	100.0	0.0	0.0	0.0	0.2	0.0	0.0	0.7
	4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Table 2 Results comparison using different models

Methods	**R-DBN**	**BDNN without marginalization**	**BDNN (99% confidence)**	**BDNN (95% confidence)**
Testing Acc	61.00%	79.07%	83.35%	73.74%
Model Uncertainty	0.6746	/	0.2344	0.1619
MI	2255	/	1459	1158

3.3 Uncertainty Test with Growing Training Data

Further experiments were also conducted to test the uncertainty of the model as the training data size changes. The following bootstrapping process is followed: 1) Split the given dataset into two subsets: a training set and a testing set. The training set includes seven of data (2000 - 2006) while the testing set includes the remaining two years of data (2007- 2008). 2) A subset of data at a specific amount of years is drawn from the training data set. 3) The subset of data of a specific size is then used to calibrate or train the candidate models, which is subsequently used to predict the collisions at the testing data set. The Acc and Mu are then calculated. 4) Repeat Step 2) to 3) for 10 times. After done, calculate the average, minimum and maximum. Results are shown in Table 3 and Figure 4.

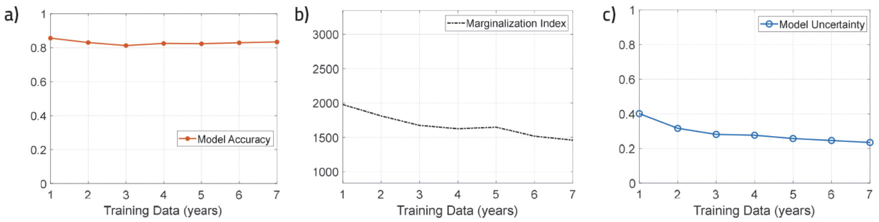

Figure 4 Effect of data size on the model

The training and testing results for the BDNN model are shown in Figure 5. When the size of the training data increases, the testing MI and Mu of the BDNN model both decrease, which is a sign of model improvement. It should be noted that the testing MI and Mu both showed an initial quick drop but then begin to level off, suggesting that the model is reaching its limit. This also suggests that BDNN learns more information using fewer data as with our previous model, the testing MAE and RMSE kept dropping dramatically as more data was added. Furthermore, figure 5(a) shows that the testing accuracy remains between 80-90 % all the time, which further suggests that the model has achieved a high level of stability.

Table 3 Testing results with increased data size

Training data (years)	1	2	3	4	5	6	7
Testing Accuracy	0.8557	0.83	0.8125	0.8249	0.8227	0.8285	0.8335
Model uncertainty	0.4012	0.3157	0.2811	0.2762	0.2569	0.2453	0.2344
Marginalization Index	1977	1811	1674	1625	1648	1517	1459

3.4 Uncertainty Test with Unknown Observation

This experiment is designed to testify the generalization and uncertainty of the BDNN. We hope once the model is well trained, it can not only perform well on the existing dataset, but also learn and act well on new or limited databases with unpredictable scenarios, and that it make continued predictions as the new information is updated, even if it has not been observed previously in its past training data sets. The following experimental process is followed. 1) Some specific datasets are carefully picked out for model training. All the datasets are in the range of specific risk levels, yet the exact collision amounts are not observed before. 2) Train a BDNN using the new designed database as in experiment 1, then test the trained model on unseen observations. 3) Repeat the process for 10 times.

In the database designed, all 305 cases that have 2 collisions (risk level 1), 102 cases with 10 collisions (risk level 2) and 27 observations that are more than 300 collisions are removed and reserved for model testing (risk level 4). Thus, during the training process, only samples with less than 300 collisions will be learned by the model. The result is shown in table 4 and 5.

Table 4 Model comparison with unseen observations

Methods	Acc	MI	Mu
DBN	14.29%	1645	0.6976
BDNN	70.83%	591	0.1632

Table 5 Model predictions

Ground truth		1	1	2	2	4	4	4	4	4
DBN predictions		2	2	3	4	3	2	3	2	2
BDNN Predictions (%)	1	13.52	99.36	0.00	0.00	0.17	0.00	0.01	0.00	0.00
	2	86.48	0.64	0.00	0.00	50.00	1.00	50.00	1.00	25.00
	3	0.00	0.00	1.00	1.00	49.83	0.00	49.99	0.00	50.00
	4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	25.00

Table 4 compares the results of different models and training methods. When compared to our previous work, substantial improvements in the accuracy, marginalization index and model uncertainty are achieved in for unseen scenarios. When using BDNN, the testing accuracy stays as high as 70.83 % in the total new dataset. Model uncertainty is only 0.1632 for the BDNN model compared to 0.6976 using DBN and 0.2344 in experiment 1. In table 5, 9 randomly selected samples are shown, with 2 samples chosen from risk level 1, 2 from risk level 2 and 5 from level 4. All the predictions by DBN are incorrect, while the predictions by BDNN are much more reasonable, especially on the five examples with risk level 4 where all the samples have over 300 collisions and not in the range of training data. Despite this, the BDNN model shows strong performance and adaptability and can still make reasonable predictions.

4 Conclusion

In this paper, we proposed an improved deep neural network approach for modelling road collisions with low uncertainty and high stability. The research has made two main contributions. First, a BDNN model is introduced as an alternative to traditional machine learning for predicting expected collision risk levels of highways. This model is able to receive and process continuous real-world data and output probabilities on each level using the techniques of Bayesian regularization, marginalization and Softmax. Second, an uncertainty analysis method is proposed, and three experiments are designed to test the model’s accuracy, marginalization index and model uncertainty. Finally, results showed that the proposed method is much more stable than our previous model and gives more reasonable suggestions to the public in real-world applications.

Acknowledgement

This research is supported by National Science and Engineering Research Council of Canada (NSERC), the Ministry of Transportation Ontario (MTO) through its Highway Infrastructure Innovation Funding Program (HIIFP), Beijing Postdoctoral Science Foundation under Grant ZZ-2019-65, Chaoyang District Postdoctoral Science Foundation under Grant 2019ZZ-45, Educational reform project of Beijing information science and technology university (2019JGYB09) and Beijing Municipal Education Commission under grant KM201811232016.

References

[1] Voosen, P.: How AI detectives are cracking open the black box of deep learning,Science, 357 (2017) 346, pp. 22-28

[2] LeCun, Y., Bengio,Y., Hinton, G.: Deep learning, Nature, 521 (2015) 7553, pp. 436-444

[3] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision, Proceedings of the IEEE Conference on Computer Vision and Pattern, pp. 2818-2826, 2016.

[4] Chollet, F: Xception: Deep Learning with Depthwise Separable Convolutions, Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1800-1807, 2016.

[5] He, K., Zhang, X., Ren, S.: Identity Mappings in Deep Residual Networks, European Conference on Computer Vision, pp. 630-645, 2016.

[6] Chen-Mccaig, Z., Hoseinnezhad, R., Bab-Hadiashar, A.: Convolutional neural networks for texture recognition using transfer learning, International Conference on Control, Automation and Infor Recognition mation Sciences, pp. 187-192, 2017.

[7] Zhang, Z., He, Q., Gao, J.: A deep learning approach for detecting traffic accidents from social media data, Transportation Research Part C: Emerging Technologies, 86 (2018), pp. 580-596

[8] Lv, Y., Duan, Y., Kang, W.: Traffic flow prediction with big data: A deep learning approach, IEEE Transportation, Intelligent Transportation Systems, 16 (2015) 2, pp. 865-873

[9] Pan, G., Fu, L., Yu, R. Muresan, M.: Winter Road Surface Condition Recognition Using a Pre-trained Deep Convolutional Neural Network, Transportation Research Board 97th Annual Meeting, Washington, D.C, 2018.

[10] Pan, G., Fu, L., Thakali, L., Muresan, M., Yu, M.: An Improved Deep Belief Network Model for Road Safety Analyses, The Transportation Research Board Annual Meeting, Washington, D.C, 2018.

[11] Muresan, M., Fu, L., Pan, G.: Adaptive Traffic Signal Control with Deep Reinforcement Learning - An Exploratory Investigation, The Transportation Research Board Annual Meeting, Washington, D.C, 2018.

[12] Pan, G., Fu, L., Thakali, L.: Development of a Global Road Safety Performance Function Using Deep Neural Networks, International Journal of Transportation Science & Technology, 6 (2017) 3, pp. 159-173

[13] Yarin, G.: Uncertainty in deep learning, Discussion, PhD thesis, University of Cambridge, 2016.

[14] Kendall, A., Roberto C.: Modelling uncertainty in deep learning for camera relocalization, IEEE international conference on Robotics and Automation (ICRA), pp. 4762-4769, 2016.

[15] Kendall, A., Yarin G.: What uncertainties do we need in Bayesian deep learning for computer vision?, Neural information processing systems, pp. 5574-5584, 2017.

[16] Lakshminarayanan, B., Alexander, P., Charles, B.: Simple and scalable predictive uncertainty estimation using deep ensembles, Neural Information Processing Systems, pp. 6402-6413, 2017.

[17] Gurevich, P., Hannes, S.: Pairing an arbitrary regressor with an artificial neural network estimating aleatoric uncertainty, Neurocomputing, 350 (2019), pp. 291-306

[18] Zhu, L., Nikolay, L.: Deep and confident prediction for time series at uber, ,2017 IEEE International Conference on Data Mining Workshops (ICDMW), pp. 103-110, 2017.

[19] Leibig, C., Vaneeda, A., Murat, S.A., Philipp, B., Siegfried, W.: Leveraging uncertainty information from deep neural networks for disease detection, Scientific reports, 1 (2017), pp. 8-16