Environmental Sensitivity Evaluation of Neural Networks in Unmanned Vehicle Perception Module

Yuru Li[1,2], Dongliang Duan[3], Chen Chen[1], Xiang Cheng[1,2], and Liuqing Yang[4]
1. State Key Laboratory of Advanced Optical Communication Systems and Networks, Department of Electronics, School of Electronics Engineering and Computer Science, Peking University, Beijing China
2. Key Laboratory of Wireless Sensor Network & Communication, Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences, Shanghai, China
3. Department of Electrical and Computer Engineering, University of Wyoming, Laramie, WY, USA
4. Department of Electrical and Computer Engineering, Colorado State University, Fort Collins, CO, USA.

***Abstract*—For autonomous driving of unmanned vehicles in intelligent transportation systems, multi-vehicle cooperative perception supported by vehicular networks can greatly improve the accuracy and reliability of the perception decisions. Currently, the perception decisions for a single vehicle are mostly provided by neural networks. Therefore, in order to fuse the perception decisions from multiple vehicles, the credibility of the neural network outputs needs to be studied. Among various factors, the environment is one of the most important affecting vehicles' perception decisions. In this paper, we propose a new evaluation criteria for the neural networks used in the perception module of unmanned vehicles. This criterion is termed as Environmental Sensitivity (ES), indicates the sensitivity of the network to environmental changes. We design an algorithm to quantitatively measure the ES value of different perception networks based on the extracted features. Experimental results show that our algorithm can well capture the sensitivity of the network in different environments and the ES values will be helpful to the subsequent decision fusion process.**

***Index Terms*—Environmental sensitivity, multi-vehicle cooperative perception, decision fusion, vehicular network**

I. Introduction

Environmental perception is an important component in autonomous driving for unmanned vehicles in the intelligent transportation system. It provides the environmental information to facilitate the decision making process for vehicles [1]. The vision-based perception method is one of the most prevalent perception schemes for unmanned vehicles [2]. It takes the camera as the main sensor and accomplishes the detection and tracking tasks mainly based on image data or video data. However, in the actual driving environments, there exist many occlusion and blind areas, as well as various extreme weather conditions. These could result in a great compromise on the perception range and accuracy, which could further lead to fatal accidents.

This work was in part supported by the Ministry National Key Research and Development Project under Grant 2017YFE0121400, Guandong Key R&D Project under Grant 2019B010153003, the open research fund of Key Laboratory of Wireless Sensor Network & Communication under Grant 2017003, Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences, and the National Science Foundation under Grants CNS-1932413 and CNS-1932139.

In order to address these problems and improve the safety of autonomous driving in practical transportation systems, many researchers proposed cooperative autonomous driving (see e.g. [3]–[5]), where vehicles interact with surrounding vehicles through the communication links among vehicles provided by the vehicular network [6]–[8]. During the perception process, each vehicle will share its own perception information with other surrounding vehicles and fuse the information collected throughout the vehicular network to obtain a more comprehensive and accurate perception outcome. In the literature, most work on multi-vehicle cooperative perception are based on the fusion of raw sensing data (see e.g. [9], [10]), which introduces heavy communication burden during the cooperation process and might be impractical for the existing vehicular networks. A more practical option is the cooperative perception at the decision level, where the vehicles only share with each other their local perception decisions, which mainly include the category and location information of obstacles in the scene.

The credibility of local decisions is a key enabler of the multi-vehicle decision fusion. In most existing researches on environmental perception of autonomous vehicles such as pedestrian detection [11], vehicle detection [12] and traffic sign detection [13], the decisions are obtained through deep learning which refers to a set of learning methods based on neural network [14]. Therefore, to obtain the credibility of the vehicles' local perception decisions, it is necessary to evaluate the output of the neural network. Compared with traditional image processing methods, the deep learning technology has many advantages in object detection and recognition: it has strong expressive and learning capability to automatically learn hierarchical features from a large number of data and optimize multiple tasks at the same time [15]. However, it is difficult to understand and explain the working principle of neural networks, because the network parameters are empirically learned from a large training dataset, lacking of rules and boundaries. This makes the networks unable to report the credibility of their outputs. Researchers have found that when the network receives some abnormal inputs or inputs that are very different from the training dataset, it often produces some unexpected outputs with high confidence [16]. Although

978-1-7281-3106-1/20/$31.00 ©2020 IEEE

by increasing the training data size and conducting iterative learning, the network errors can be reduced, in such an open application scenario as the autonomous driving with a variety of environments and unexpected inputs, decision errors would almost be unavoidable. Meanwhile, a minor error of the environmental perception network may lead to fatal accidents. An example is the well-known Tesla autonomous driving accident, which is due to the failure to detect a truck because of the lighting condition. Therefore, given the potential errors in the reported decisions, it is crucial to provide a scheme for the neural networks to report quantified credibility on their outputs.

As illustrated in the Tesla and several similar accidents, environment such as lighting condition is a key factor that affects the network performance for autonomous driving of unmanned vehicles. Intuitively, when an unmanned vehicle drives in an environment similar to a scene encountered during the training process, the perception result should have a high credibility, and when the driving environment changes and is quite different from all the scenes in the training set, the output of the network should have a low credibility. This is somewhat similar to human drivers. However, the difference is that, though human drivers may also make wrong decisions under abnormal conditions, at the same time they will realize that they have not encountered such environment before and their perceptions under such environment are less credible and they would make adjustments accordingly such as reducing speed or suspending driving. Therefore, we say that human drivers have *environmental sensitivity*. However, the existing neural networks adopted by unmanned vehicles do not necessarily have this capability. In order to study the credibility of network outputs, it is necessary to first evaluate their ES.

The environmental changes that we focus on mainly refers to the natural influences on the environment such as weather and lighting conditions rather than simple scene changes. This is because during driving the scene changes are usually slow, while in a set of continuous scenes, the environmental changes are often sudden such as sudden strong light, a rainstorm, or dusty weather. These environmental changes may cause the perception network to fail. If the network cannot even realize that the current input is abnormal, the vehicle would not know whether and how to take advantage of the multi-vehicle cooperation. In other words, the premise of obtaining the output credibility of the network is that the network has a strong capability to perceive the changes of the environment, i.e., when the input image data is influenced by some natural conditions, the network is capable of recognizing the abnormality, so as to give its own output a low credibility. Then, the vehicle can seek cooperation from surrounding vehicles to improve or correct its perception decisions.

In this paper, in the context of the multi-vehicle cooperative perception in the vehicular network, we focus on the perception network for image processing and propose a new network evaluation criteria, namely the Environmental Sensitivity (ES), which evaluates the perception capability of a neural network to environmental changes. Basically, we want to evaluate how similar the images obtained from the same scene are under different environments for neural networks. It should be noted that the similarity as perceived by neural networks is quite different from those by human beings. Images that are judged to be similar by humans are not necessarily to be perceived as similar by neural networks, as illustrated in the adversarial examples [17]. Therefore, in order to capture the interpretation of the images by neural networks, our similarity measure or equivalently distance measure is based on the features extracted by the network, which determines the final outputs provided by the neural networks. By studying the distribution of image features collected from different environments extracted by the network, we combine the inter-dispersion and intra-dispersion to obtain the ES value. While network accuracy has been widely adopted to evaluate the performance of neural networks, the ES proposed in this paper is another important evaluation criteria of neural networks and pays more attention to the security and reliability of the network. Based on this value, one can further model the credibility of network output, which can be used in the fusion process of multi-vehicle cooperative perception.

The rest of this paper is organized as follows. In Section II, we present some related researches about neural network evaluation and their limitations. In Section III, we introduce our algorithm to quantify the ES. In Section IV, we use the actual driving image data to verify our algorithm, and apply the ES evaluation algorithm to four classical feature extraction networks commonly adopted for detection tasks. Finally, we summarize our work and analyze the significance of the idea proposed in this paper to the follow-up work in Section V.

II. Related Work

There are mainly two methods to evaluate the sensitivity of neural networks in the literature. One is to find the lower bounds of input data changes that are required to cause an error in the network output [18], [19]. However, the computation complexity of this method is very high. It is often based on the estimation of closed-form solutions or experimented on small-scale networks and is difficult to be extended to the large-scale complex perception network for autonomous driving.

The other sensitivity evaluation method is testing based, which tries to find the network bugs by generating a number of testing examples. During this process, the generation of effective testing examples is the key issue. Two common practices are adversarial attack and neuron coverage. Adversarial attack based technique [20], [21] is to generate adversarial examples [17] that attack the network and use the attack success rate or minimum distortion of input to evaluate the network. For a neural network, the easier it is to build an adversarial example, the less robust and more sensitive the network is. Neuron coverage based technique [22], [23] is to generate bad test inputs by maximizing the neuron coverage [22] of network and evaluate the sensitivity of the network according to the testing. The larger the neuron coverage is, the more effective the testing process is. Based on the neural coverage, some improved coverage based techniques are also proposed [24].

For the neural networks used in the perception module of unmanned vehicles, the existing evaluation methods have two major issues. The first is the lack of specific context. Most researches on network evaluation rarely consider the application scenario. They evaluate the network on several arbitrarily selected applications, while no one can guarantee that their approach will work for all scenarios. Autonomous driving is a special scenario which contains more complex and diverse environments. So its requirements for the stability and reliability of the perception network are higher. Therefore, more targeted evaluation methods for unmanned vehicle perception module are needed. The second and more important issue is that the evaluation criteria are not comprehensive. The main concern of the existing network sensitive evaluation methods is the range of inputs that a network can handle correctly, which is certainly important, but not sufficient. In autonomous driving, it is one thing for the perception network to handle as many input types as possible, yet it is another thing for the network to deal with different environmental types. Hence, one needs to designate another measure to evaluate a network's sensitivity to the driving environment.

In this paper, we focus on the perception network for automated driving and propose a new evaluation criteria, which measures the capability of the network to perceive environmental changes. Only when the network can distinguish environments with small differences into distinct classes, it may have perception ability to recognize the abnormal environments that do not exist in the training dataset. ES is another evaluation criteria different from the accuracy. It concerns more on the security and reliability of the network.

III. ES

ES measures the network's capability to distinguish environmental changes. The environmental changes that we mainly focus on is the influences on the perception network imposed by natural conditions. That is, by creating distortions to the original images under different natural influences, we evaluate the changes in the perception results of the network. We define the images from the same environment as several photos taken continuously by the camera under the same natural influences.

The ES value is calculated based on the features extracted from the convolution layer in the perception network. On the one hand, this is because feature extraction is a very important step for the perception network and the subsequent classification and regression tasks are based on these features. On the other hand, the distance measures to distinguish input images as perceived by neural networks is quite different from those by human beings. While the human beings judge on the original images, neural networks do not. Taking the instance of adversarial examples [17], the small difference that is hard to be detected by human will be considered as two completely different inputs by the neural networks. Therefore, features can be regarded as the abstract representation of the original images perceived by the network.

ES includes two important factors: one is the inter-dispersion, which indicates how disperse are the features of different environments. The other is the intra-dispersion, which indicates how dense are the features of the same environment. The higher the feature dispersion of different environments, the better the ability to distinguish different environments of the network. The lower the feature dispersion of the same environment, the better the environmental information extracted by the network. Therefore, the higher the inter-dispersion and the lower the intra-dispersion, the higher the ES value. Fig. 1 is a 2D visualization example of the features under two different environments. It can be seen that the features of these two environments are well differentiated and the image features from the same environment are relatively concentrated, which shows that the network is sensitivity to these two environments. In order to quantify the ES, we consider the intra-dispersion and inter-dispersion respectively.

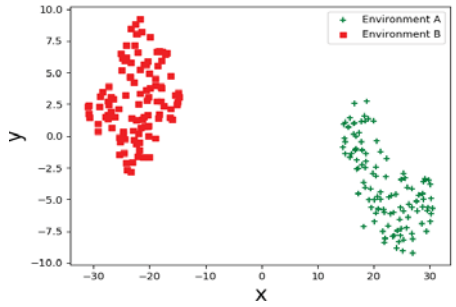


Fig. 1. The distribution of features extracted from MobileNet-SSD.

Suppose there are K kinds of environments in total, which are $\{C_1, C_2, \ldots, C_K\}$, the number of features of each class is denoted as $\{n_1, n_2, \ldots, n_K\}$. Then, the total feature matrix can be expressed as $\boldsymbol{F}_{d\times n}$ where d is the dimension of features and $n = n_1 + n_2 + \ldots + n_K$ is the total number of features.

In the distribution of environmental features, there might be some abnormal features, which deviate from most features. These abnormal features have an impact on the mean value of all features. Therefore, we should consider the weighted average value instead of the mean value of the features, where the weight of each data is determined by the abnormal degree of features. Suppose the data set whose mean value is to be calculated is $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m\}$. For data $\boldsymbol{x}_i$, its k-nearest neighbor $\boldsymbol{x}_j$ can be expressed as $\boldsymbol{x}_j = \text{Knn}(\boldsymbol{x}_i)$ where k is usually a small value, and its k-nearest neighbor distance represents the distance between $\boldsymbol{x}_i$ and $\text{Knn}(\boldsymbol{x}_i)$, expressed in Eq. (1). The distance metric we use here is Euclidean distance, denoted by $\boldsymbol{L}_2$.

$$D_k(\boldsymbol{x}_i) = \boldsymbol{L}_2(\boldsymbol{x}_i, \text{Knn}(\boldsymbol{x}_i)) = \boldsymbol{L}_2(\boldsymbol{x}_i, \boldsymbol{x}_j) \ , \tag{1}$$

the weight of $\boldsymbol{x}_i$ is calculated as follows:

$$\alpha_i = \frac{D_k\left(\boldsymbol{x}_i\right)}{D_k\left(\text{Knn}(\boldsymbol{x}_i)\right)} = \frac{D_k\left(\boldsymbol{x}_i\right)}{D_k\left(\boldsymbol{x}_j\right)} \ , \tag{2}$$

$$\rho_i = \max\left(\alpha_i, 1\right) \ , \tag{3}$$

$$w_i = e^{(-\lambda(\rho_i - 1))} \ , \tag{4}$$

where α_i in Eq. (2) is the ratio of k-nearest distance of data $\boldsymbol{x}_i$ to that of data $\boldsymbol{x}_j = \text{Knn}(\boldsymbol{x}_i)$. If $\alpha_i > 1$, it indicates that the density near $\boldsymbol{x}_i$ is less than that near $\boldsymbol{x}_j$, so $\boldsymbol{x}_i$ may be an outlier. If $\alpha_i \leq 1$, it indicates that the density near $\boldsymbol{x}_i$ is greater than or equal to that near $\boldsymbol{x}_j$, so $\boldsymbol{x}_i$ is a dense point with respect to $\boldsymbol{x}_j$. Hence, we define the outlier factor ρ_i in Eq. (3) as the maximum value of α_i and 1. If $\rho_i = 1$, $\boldsymbol{x}_i$ is not an outlier. Otherwise, the larger the ρ_i, the greater the outlier degree of data $\boldsymbol{x}_i$. In Eq. (4), λ is an adjustment parameter in the exponential function, based on which, ρ_i is mapped to the weight of data $\boldsymbol{x}_i$, w_i, with range from zero to one.

Given the weights of all data in X, the weighted average value can be obtained as:

$$\boldsymbol{m} = \frac{\sum_{i=1}^{n} w_i \boldsymbol{x}_i}{\sum_{i=1}^{n} w_i} \ . \tag{5}$$

Assuming the weight average vector of features in class C_j is $\boldsymbol{m}_j$, then the intra-dispersion matrix of class C_j can be expressed as:

$$\boldsymbol{S}_j = \sum_{\boldsymbol{f}_i \in C_j} (\boldsymbol{f}_i - \boldsymbol{m}_j)(\boldsymbol{f}_i - \boldsymbol{m}_j)^T \ , \tag{6}$$

where $\boldsymbol{S}_j$ is a symmetric matrix, similar to the covariance matrix, but without the expectation operation. The total intra-dispersion matrix is the sum of all classes' intra-dispersion matrix, expressed as follows:

$$\boldsymbol{S}_{\text{intra}} = \sum_{j=1}^{K} \boldsymbol{S}_j \ . \tag{7}$$

The inter-dispersion measures the distribution of features among different classes. Assuming that $\boldsymbol{m}$ represents the mean vector of all features, the inter-class dispersion matrix can be expressed as:

$$\boldsymbol{S}_{\text{inter}} = \sum_{j=1}^{K} (\boldsymbol{m}_j - \boldsymbol{m})(\boldsymbol{m}_j - \boldsymbol{m})^T \ . \tag{8}$$

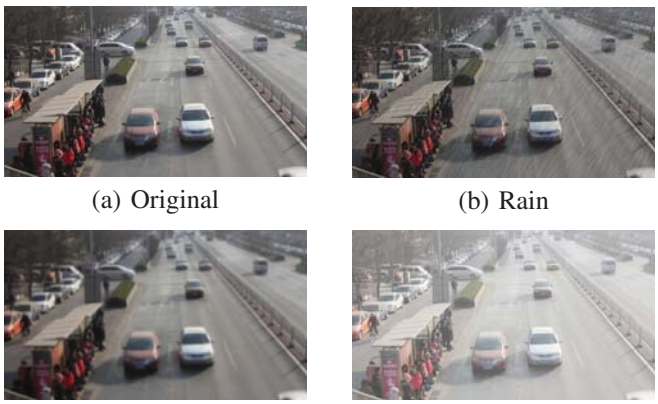
The inter-dispersion matrix does not consider the inner data distribution, but uses the weighted average vectors to represent all features in the corresponding class and measures the distribution of these weighted average vectors.

ES is the relative ratio between the inter-dispersion and the intra-dispersion. For now, the dispersions are given in matrices. In order to convert the matrix into scalar, we take the trace of the matrix to represent the dispersion. This is because both the intra-dispersion matrix and inter-dispersion matrix are symmetric. The trace of the symmetric matrix is equal to the sum of the eigenvalues, which can represent the dispersion of the data in the projection direction of the eigenvector. So the sum of eigenvalues can reflect the dispersion degree of data distribution. As a result, ES can be expressed as:

$$ES = \frac{\text{tr}(\boldsymbol{S}_{\text{inter}})}{\text{tr}(\boldsymbol{S}_{\text{intra}})} \ . \tag{9}$$

IV. Experiments

To illustrate the proposed measure, we conducted experiments using the UA DETRAC data set [25], [26] which consists of 10 hours of videos captured by a Cannon EOS 550D camera at 24 different locations at Beijing and Tianjin in China. We assume that 100 consecutive images collected by the camera are in the same scene. Based on the images in the same scene, we simulate three different environments by adding distortions due to different natural influences, including rain, fog and strong light, and take the original images and the three naturally distorted images as four different environments to evaluate the distribution of features extracted by the network. Fig. 2 gives an example on the images with different distortions under different environments.



(a) Original (b) Rain

(c) Fog (d) Strong light

Fig. 2. The original image and the images under different environment conditions.

High-dimensional features are usually so sparse that it is difficult for the model to find the relations between features, and the computational complexity of calculating the distances among features is very high. So we use t-SNE (t-Distributed Stochastic Neighbor Embedding) [27] to reduce the high-dimensional features to two dimensions and carry out the visualization and ES calculation in the two-dimensional space. t-SNE is a nonlinear dimensionality reduction algorithm for exploring high-dimensional data. It converts the high-dimensional Euclidean distance into a conditional probability distribution representing similarity, and constructs the probability distribution of these points in the low-dimensional space to make the two probability distributions as similar as possible. The advantage of t-SNE is that it can maintain the local structure of the high-dimensional data, i.e., the points with similar distance in the high-dimensional space are still similar in the low-dimensional space. It is commonly used in high-dimensional data visualization. The dimensionality reduction process is similar to the data compression process, making the feature distribution more compact while maintaining the relative relationship between features as much as possible.

For comparative studies to illustrate our proposed ES measure, we select four classic feature extraction networks:

VGG16 [28], VGG19 [28], Inception-V3 [29] and ResNet50 [30], which are all pre-trained on ImageNet. It is a reasonable practice because for most classification or detection tasks, the convolution layers are usually fine tuned on the basis of these pre-trained feature extraction networks.

We randomly select 200 scenes from the dataset, equivalent to 20,000 original and 60,000 distorted images and evaluate the ES of different networks for every scene based on these images. We finally obtain the $\boldsymbol{ES}$ matrix with the size of 800×4. Because the value ranges of features in different scenes are quite different, we normalized the ES values of different networks in each scene using Z-score standardized method [31], which can convert data of different magnitude into uniform Z-score scores for comparison. The conversion formula is as follows:

$$es_{ij}^{*} = \frac{es_{ij} - \mu_i}{\sigma_i} \ , \tag{10}$$

where $i \in [1, 800]$ is the index of the scene, $j \in \{1, 2, 3, 4\}$ is the index of the corresponding network, es_{ij} is the original ES value of the network j in the i-th scene, μ_i and σ_i are the mean ES value and standard deviation value of different networks in the i-th scene respectively, and es_{ij}^{*} is the normalized ES value. After the normalization, we can obtained the normalized ES matrix $\boldsymbol{ES}^{*}$.

Fig. 3 shows the distribution of features extracted by four networks in three selected scenes. Different rows represent different scenes, and four figures in each row represent the distribution of features extracted by the four networks in the same scene. The title of each figure gives the name of the network generating the features and the ES value calculated based on the feature distribution. It can be seen that the performance of Inception-V3 is the worst in the three scenes. The features of different environments are overlapped together and cannot be distinguished well. The features extracted by VGG16 can be distinguished easily in scene (a) and scene (b). However, in scene (c) the features of three environments are mixed together and the distribution are very dispersed in each environment. As for VGG19, its ES value is high in scene (a), but the features in scene (b) and scene (c) are too dispersed, resulting in the lower ES values. By comparison, ResNet 50 has the best performance in ES evaluation process based on these three scenes. Its features are centralized within the class and dispersed sufficiently among classes in scene (a) and (b). In scene (c), though the features in rainy days are very close to the original features, the distribution is still better than the other three networks in the same scene with the highest ES value.

In order to reflect the average ES level of each network, we calculate the mean of ES values for every network respectively based on the $\boldsymbol{ES}^{*}$, which is the result of all 200 scenes studied. The results are presented in Table I. It can be seen that the order of the four networks based on the average ES values from high to low is ResNet50 $>$ VGG16 $>$ VGG19 $>$ Inception-V3, which is basically consistent with the analysis based on Fig. 3. This shows that our ES evaluation algorithm can well reflect the sensitivity of network to the environmental changes based on the corresponding feature distributions.

TABLE I
MEAN NORMALIZED ES VALUES OF FOUR NETWORKS

	VGG16	***VGG19***	***Inception-V3***	***ResNet50***
mean ES	***-0.09317394***	***-0.24366768***	***-0.9122827***	***1.24912431***

V. CONCLUSIONS

This paper presented a new neural network evaluation criterion, namely the Environmental Sensitivity (ES), which aims at the neural networks used in the perception module of autonomous vehicles. This criterion is different from the commonly adopted one, the accuracy. It concerns more about the safety and reliability of the network, and is used to evaluate the capability of the network to capture various environmental changes. We designed the ES evaluation algorithm by measuring the distribution of the features extracted by the neural networks under different environments. In the experiments, we used the algorithm to evaluate four commonly used feature extraction networks and results showed that our algorithm can well capture the sensitivity of the network to environmental changes. Based on the work of this paper, according to a network's ES value, one can further model the credibility of the network outputs, that is, the reliability of the perception decision for a single vehicle. This can be used in the decision fusion process of multiple vehicles in the vehicular network so as to improve each vehicle's perception performance.

REFERENCES

[1] L. Li, K. Ota, and M. Dong, "Humanlike driving: Empirical decision-making system for autonomous vehicles," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 8, pp. 6814–6823, Aug 2018.

[2] S. Sivaraman and M. M. Trivedi, "Looking at vehicles on the road: A survey of vision-based vehicle detection, tracking, and behavior analysis," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 4, pp. 1773–1795, 2013.

[3] S.-W. Kim, W. Liu, M. H. Ang, E. Frazzoli, and D. Rus, "The impact of cooperative perception on decision making and planning of autonomous vehicles," *IEEE Intelligent Transportation Systems Magazine*, vol. 7, no. 3, pp. 39–50, 2015.

[4] R. Hult, G. R. Campos, E. Steinmetz, L. Hammarstrand, P. Falcone, and H. Wymeersch, "Coordination of cooperative autonomous vehicles: Toward safer and more efficient road transportation," *IEEE Signal Processing Magazine*, vol. 33, no. 6, pp. 74–84, 2016.

[5] X. Cheng, D. Duan, L. Yang, and N. Zheng, "Cooperative intelligence for autonomous driving," *ZTE Communications*, 2019.

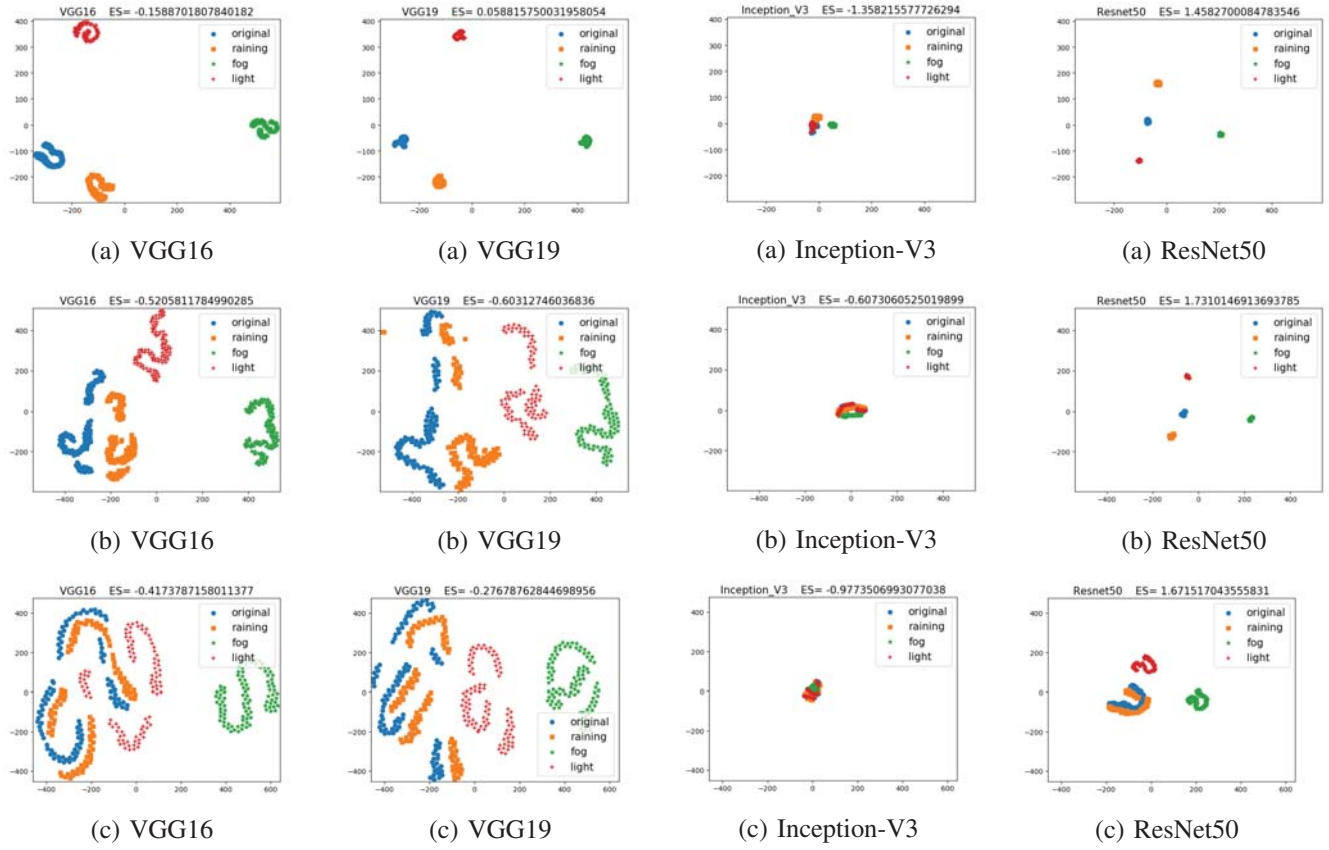
[6] X. Cheng, C. Chen, W. Zhang, and Y. Yang, "5G-enabled cooperative intelligent vehicular (5GenIV) framework: When Benz meets Marconi," *IEEE Intelligent Systems*, vol. 32, no. 3, pp. 53–59, May/June 2017.

[7] X. Cheng, R. Zhang, and L. Yang, "Wireless towards the era of intelligent vehicles," *IEEE Internet of Things Journal*, vol. 6, no. 1, pp. 188–202, February 2018.

[8] ——, *5G-enabled vehicular communications and networking,* Springer, Cham, Switzerland, 2018.

[9] H. Li, M. Tsukada, F. Nashashibi, and M. Parent, "Multivehicle cooperative local mapping: A methodology based on occupancy grid map merging," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 5, pp. 2089–2100, 2014.

[10] Y. Li, D. Duan, C. Chen, X. Cheng, and L. Yang, "Occupancy grid map formation and fusion in cooperative autonomous vehicle sensing," in *2018 IEEE International Conference on Communication Systems (ICCS)*. IEEE, 2018, pp. 204–209.



(a) VGG16 (a) VGG19 (a) Inception-V3 (a) ResNet50

(b) VGG16 (b) VGG19 (b) Inception-V3 (b) ResNet50

(c) VGG16 (c) VGG19 (c) Inception-V3 (c) ResNet50

Fig. 3. The distribution of features extracted by four networks in three different scenes.

[11] J. Li, X. Liang, S. Shen, T. Xu, J. Feng, and S. Yan, "Scale-aware fast r-cnn for pedestrian detection," *IEEE Transactions on Multimedia*, vol. 20, no. 4, pp. 985–996, 2017.

[12] X. Hu, X. Xu, Y. Xiao, H. Chen, S. He, J. Qin, and P.-A. Heng, "Sinet: A scale-insensitive convolutional neural network for fast vehicle detection," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 3, pp. 1010–1019, 2018.

[13] Y. Zhu, C. Zhang, D. Zhou, X. Wang, X. Bai, and W. Liu, "Traffic sign detection and recognition using fully convolutional network guided proposals," *Neurocomputing*, vol. 214, pp. 758–766, 2016.

[14] M. A. Nielsen, *Neural networks and deep learning*. Determination press San Francisco, CA, USA:, 2015, vol. 25.

[15] Z.-Q. Zhao, P. Zheng, S.-t. Xu, and X. Wu, "Object detection with deep learning: A review," *IEEE transactions on neural networks and learning systems*, 2019.

[16] A. Nguyen, J. Yosinski, and J. Clune, "Deep neural networks are easily fooled: High confidence predictions for unrecognizable images," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 427–436.

[17] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *arXiv preprint arXiv:1312.6199*, 2013.

[18] T.-W. Weng, H. Zhang, P.-Y. Chen, J. Yi, D. Su, Y. Gao, C.-J. Hsieh, and L. Daniel, "Evaluating the robustness of neural networks: An extreme value theory approach," *arXiv preprint arXiv:1801.10578*, 2018.

[19] M. Hein and M. Andriushchenko, "Formal guarantees on the robustness of a classifier against adversarial manipulation," in *Advances in Neural Information Processing Systems*, 2017, pp. 2266–2276.

[20] S. Gu and L. Rigazio, "Towards deep neural network architectures robust to adversarial examples," *arXiv preprint arXiv:1412.5068*, 2014.

[21] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *2017 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2017, pp. 39–57.

[22] K. Pei, Y. Cao, J. Yang, and S. Jana, "Deepxplore: Automated whitebox testing of deep learning systems," in *Proceedings of the 26th Symposium on Operating Systems Principles*. ACM, 2017, pp. 1–18.

[23] Y. Tian, K. Pei, S. Jana, and B. Ray, "Deeptest: Automated testing of deep-neural-network-driven autonomous cars," in *Proceedings of the 40th international conference on software engineering*. ACM, 2018, pp. 303–314.

[24] J. Kim, R. Feldt, and S. Yoo, "Guiding deep learning system testing using surprise adequacy," in *Proceedings of the 41st International Conference on Software Engineering*. IEEE Press, 2019, pp. 1039–1049.

[25] L. Wen, D. Du, Z. Cai, Z. Lei, M. Chang, H. Qi, J. Lim, M. Yang, and S. Lyu, "UA-DETRAC: A new benchmark and protocol for multi-object detection and tracking," *arXiv CoRR*, vol. abs/1511.04136, 2015.

[26] S. Lyu, M.-C. Chang, D. Du, L. Wen, H. Qi, Y. Li, Y. Wei, L. Ke, T. Hu, M. Del Coco *et al.*, "Ua-detrac 2017: Report of avss2017 & iwt4s challenge on advanced traffic monitoring," in *Advanced Video and Signal Based Surveillance (AVSS), 2017 14th IEEE International Conference on*. IEEE, 2017, pp. 1–7.

[27] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of machine learning research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

[28] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[29] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.

[30] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[31] "Z-score: Definition, formula and calculation," https://www.statisticshowto.datasciencecentral.com/probability-and-statistics/z-score/.