Detection of agglomerate fog based on a shallow convolutional neural network

Linlin Li¹ · Bo Yang² · Shaohui Chen³

Received: 21 September 2020 / Revised: 15 August 2021 / Accepted: 9 September 2021 /

Published online: 6 November 2021

© The Author(s) 2021

Abstract

As a kind of frequent bad weather, Agglomerate fog is a serious danger to people's safe driving, especially on the highway. Therefore, the research on the detection of fog is of great practical significance to ensure the safety of pedestrians. This paper proposes a shallow convolutional neural network for agglomerate fog detection in images, including the framework of the network and the detailed design of each component. Firstly, the image is divided into several sub-images; and then a shallow convolutional neural network is constructed and employed to identify the existence of fog for each of the sub-area images; lastly, the decision results of each sub-area images were integrated to determine whether the whole image contained agglomerate fog. A large quantity of simulation data and real data were used to test the performance of the proposed method, the experimental results show that the presented method can achieve more than 90% detection accuracy, which demonstrated that the advantage of the proposed method comparing with several existed methods.

Keywords Convolutional neural network · Agglomerate fog detection · Image classification

✉ Bo Yang
13910700045@139.com

Linlin Li
xinxi_li@tpri.org.cn

Shaohui Chen
chensh01@ehualu.com

¹ School of Information Resources Management, Renmin University of China, Beijing 100872, China

² School of Emergency Technology and Management, North China Institute of Science & Technology, Langfang 065201, China

³ Beijing Gaocheng Technology Development Co. LTD, Beijing 100043, China

1 Introduction

Agglomerate fog is generally found in low-lying areas with high air humidity, which is closely related to local microclimatic environment. Agglomerate fog not only brings great safety hazard to the travel of vehicles, but also makes it extremely difficult to deal with rapid emergency response required by such disasters. With the development of computer vision technology, how to use existing roadside surveillance video to detect agglomerate fog hazard has received extensive attention from scholars. Presently, many agglomerate fog detection methods were constructed based on analyzing the characteristics of images [10, 20, 25, 27]. Also, there were many fog detection approaches were developed through machine learning [23]. The former approaches designed the corresponding methods by analyzing the influence of agglomerate fog on objects in the scene and to detect the agglomerate fog. The latter approach trains classifiers through artificial features to identify agglomerate fog. The key issue of these two kinds of methods lies in the construction of features and the design of classifiers. And both two kinds of agglomerate fog detection methods have been greatly improved in terms of accuracy.

In recent years, deep learning theory and methods have been greatly developed. Convolutional neural networks (CNN) have been widely used in image processing and analysis [35], such as image classification, object detection [36], image enhancement, and so on. At present, image de-fogging algorithms based on convolutional neural networks are common. Cai proposed a trainable end-to-end system called DehazingNet to estimate the transmittance of image blocks. The system uses the deep structure of a convolutional neural network with image blocks as the input and their transmittance as the output [2]. Ren proposed a multi-scale convolutional neural network (MSCNN). The whole network includes coarse-scale networks and fine-scale networks. The coarse-scale network is used for rough estimation of transmittance, and the fine-scale network is used for refinement of coarse transmittance estimation [24]. Li introduced the network model AODNet for image de-fogging. Unlike other methods, AODNet does not need to separately estimate the transmittance or atmospheric light but rather can obtain fog-free images directly through the network output [12].

In the study of image defogging, the first step is to detect whether there is agglomerate fog in the image. If the fogging algorithm is applied to the image without agglomerate fog, the image will become bad and the computing resource will be wasted. This paper proposes a shallow convolutional neural network model for agglomerate fog detection based on image. Comparing with the existed methods, the network layer designed in this paper is shallow, which can save computing power. At the same time, due to the block strategy and convolution layer size from large to small design ideas, taking into account the overall characteristics and local characteristics, it can achieve good detection results.

2 Related works

Fog detection can be regarded as a binary classification problem, traditional machine learning algorithms have many shortcomings for this application. For instance, they need to design features such as local binary pattern (LBP) [21], histogram of oriented gradients (HOG) [4], speeded-up robust features (SURF) [1], and so on.

Image classification algorithm based on deep learning is an end-to-end training process. The input of the neural network is an image, and the output is each category's probability. The category with the highest probability is used as the prediction category. Representation features can be extracted from the hidden layers, the process from shallow to deep extracts common features to abstract features, and the output features represent the categories.

Theoretically speaking, as long as the network layers are wide and deep enough, the neural network can fit any function. However, when dealing with complex tasks, the fully connected traditional neural networks become more complicated and the number of network parameters increases as the number of neurons becomes larger, which will lead to overfitting problem. The convolutional neural network uses a convolution operation to traverse all points, which, in essence, is a filtering kernel, and the convolution operation equivalent to the image filtering. The patterns learned by convolutional neural networks are translationally invariant, hence, the convolutional neural network can identify this pattern anywhere in an image after learning a certain pattern. Moreover, the convolutional neural network obtains patterns with spatial hierarchy, which means that the shallow convolution layer learned edge features, the deep convolutional layer learns about the features of contours combining the edges, and the last layer of the network learns about the more essential features combining the previous features.

As early as 1998, LeCun proposed the convolutional neural network LeNet [11] for handwritten character identification. A schematic diagram of the network is shown in Fig. 1, including two convolution layers, two downsampling layers, and two fully connected layers. Such a convolution–downsampling–fully-connected pattern or the like is still used in new convolutional neural networks proposed in recent years such as VGGNet [22, 28] and Inception [29–31]. LeNet has been applied to bank check identification and has achieved good results.

In 2012, Krizhevsky used AlexNet [7] to win the ImageNet Large Scale Visual Recognition Challenge (ILSVRC). The AlexNet network includes five convolution layers, three maximum pooling layers, and two local normalized layers. Compared to LeNet, AlexNet's design ideas are similar. The differences are that AlexNet uses Relu as an activation function instead of Sigmoid, and AlexNet uses a variety of methods to prevent overfitting, such as data expansion, dropout, and so on.

In 2014, a researcher at VGG Labs in Oxford University won the ILSVRC with VGGNet. This network also used convolution–pooling–full connectivity. Unlike AlexNet, VGGNet explored the depth of the network, proposed a variety of deep networks from layer 11 to 19, and achieved good results on layer 16/19. Many networks now use VGGNet as a benchmark for comparison.

Fig. 1 LeNet network structure

In recent years, such structures as Inception [29] and ResNet [5] have achieved better performance in object identification. The structure of Inception combines the features of multi-scale receptive fields, and the extracted features are richer. The residual in ResNet combines shallow features with deep features to make gradients easier to propagate backwards, which is beneficial for designing deeper networks. Subsequent improvements have been made to Inception [30, 31], ResNet [34], and the combination of Inception and ResNet [32]. Inception's and Residual's structures are shown in Fig. 2.

With the rapid development of CNN, there are many studies were carried out in many fields such as video object segmentation [15, 17, 18], object tracking [14, 16] and so on.

Focusing on fog detection from images, Bin et.al proposed a so called CNN-RNN based multi-label classification method [38]. The convolutional neural network (CNN) was extended with a channel-wise attention model to extract the most correlated visual features at first and then the Recurrent Neural Network (RNN) was adopted to process the features and excavated the dependencies among weather classes. Toward weather condition recognition and based on the understanding of the importance of regional cues, a deep learning framework named region selection and concurrency model (RSCM) was presented to discover regional properties and concurrency [13]. Face to the applications of traffic management and control, Yu proposed a Global Similarity Local-Saliency Network (GSLNet) for traffic weather recognition. This work also designed the corresponding strategy to restrict the network focusing on road weather details [37].

The above works have achieved better detection results than the non-deep learning methods, but it can also be seen that the above methods were designed for a variety of weather, and the pertinence of haze detection is not strong, so there is still room to improve the accuracy of haze detection. Considering the requirements including low time consuming and effectiveness of the haze detection algorithm, a shallow neural network considering the global and local features as well as multi-scale features of the image is designed, so as to achieve the purpose of fast and high-precision fog detection.

Fig. 2 Inception and Residual structures

3 Method

3.1 Overall framework of agglomerate fog-containing image detection algorithm

The detection of agglomerate fog-containing images can be viewed as the classification of input images into fog-containing and fog-free categories. It is not likely for all areas of an image to contain fog. In other word, it may affect the judgment result of the whole image because there is no fog in some areas of the scene. Hence, the strategy of dividing images into blocks is adopted to determine whether the whole image contains fog. The framework of the entire algorithm is shown in Fig. 3.

To divide an image, for the sake of simplicity, this paper adopts an equal splitting method as shown in Fig. 4. The proposed convolutional neural network is used to determine if a sub-area contains fog. Finally, the identification results of all sub-areas are voted, and the voting result is used as a basis to determine whether this image belongs to the fog-containing or fog-free category.

For the problem of how many sub images an image is divided into, it depends on the size of the observation field. The larger the field of view is, the more the number of sub image segmentation is. On the contrary, the less the number of sub image segmentation is. For the traffic monitoring scene, the general field of view is not too large, so it is enough to divide the image into nine sub images. For this, we give the test in the experimental part.

Fig. 3 Comprehensive determination based on sub-area-based learning

Fig. 4 Diagram of dividing an image into sub-areas

3.2 Network structure of agglomerate fog detection

The networks mentioned above, such as VGGNet, InceptionNet, and ResNet, are mostly applied in thousands of classifications, so a very deep network layer is needed to accurately extract features of different objects. For fog-containing or fog-free image classification, the task is relatively simple and does not require a very deep network. In fact, the use of a deep network for this task may lead to over-fitting. Based on this understanding, this paper proposed a shallow network for fog detection is shown in Fig. 5.

As shown in Fig. 5, the proposed network consists of the following parts:

1. Convolution layer: the convolution kernel size is 7×7 , the number of convolution kernels is 96, and the step size is 1.
2. Maximum pooling layer: the pooling range is 2×2 , and the step size is 2.
3. Convolution layer: the convolution kernel size is 5×5 , the number of convolution kernels is 256, and the step size is 1.
4. Maximum pooling layer: the pooling range is 2×2 , and the step size is 2.

Fig. 5 Network structure of fog detection

5. Convolution layer: the convolution kernel size is 3×3 , the number of convolution kernels is 384, and the step size is 1.
6. Maximum pooling layer: the pooling range is 2×2 , and the step size is 2.
7. Fully connected layer: the number of neurons is 512.
8. Fully connected layer: the number of neurons is 512.
9. Softmax layer: the number of categories is 2.

The size of the convolution kernel is gradually reduced from 7×7 to 3×3 , it is benefit to the shallow layer extracts the information of a large field of view and the deeper layer extracts the finer information. The first six convolutional pooling layers are used for feature extraction, and the fully connected layer and the Softmax layer can be regarded as classifiers. By sending the multi-scale features extracted from the first six layers to the later classifiers, the detection of foggy images is carried out. It can be seen that the convolution network designed here has fewer layers and takes into account the characteristics of different scales, so it can help to realize the detection of fog image quickly and accurately, this is just the advantage of the proposed method.

3.3 Feature extraction

The network performs feature extraction through a convolution pooling operation. The convolution operation completes the learning of different patterns, and the pooling operation completes down-sampling and extracts obvious responses.

The convolution operation is performed by traversing the filter kernel in a sliding window and applying the convolution kernel to the neighboring image block around each point. Taking a two-dimensional image I as an input, for a two-dimensional convolution kernel K , the convolution result is shown in Eq. (1).

$$S(i, j) = (I * K)(i, j) = \sum_m \sum_n I(m, n) K(i - m, j - n) \quad (1)$$

Since the convolution is commutative, Eq. (1) can be rewritten as Eq. (2).

$$S(i, j) = (K * I)(i, j) = \sum_m \sum_n I(i - m, j - n) K(m, n) \quad (2)$$

The cross-correlation operation is almost the same as the convolution operation, but the kernel is not flipped, as shown in Eq. (3)

$$S(i, j) = (I * K)(i, j) = \sum_m \sum_n I(i + m, i + n) K(m, n) \quad (3)$$

It should be noted that convolution operations in many machine learning libraries are replaced by cross-correlation functions, which are also known as convolutions. The convolutional layers mentioned in the previous section are all cross-correlation operations. The reason is that the two operations can achieve the same result. Even if the convolution operation is used, the weights obtained through training can be flipped to obtain the same weight as the cross-correlation operation. The cross-correlation can eliminate the operation of kernel flip.

There are many types of pooling, including average pooling and maximum pooling, which are mean filtering and maximum filtering. The maximum pooling operation obtains

the largest response in a neighborhood and can also be down-sampled by setting the step size. If the step size is set to two, the width and height are sampled to half of the original. When the input makes a small amount of translation, the pooling operation can make the representation of the input approximately unchanged.

3.4 Activation function

In order to make the deep neural network nonlinear, a nonlinear activation function needs to be added to the output of the neuron. If there is no nonlinear activation function (only linear combination), the deep network is equivalent to a single-layer perceptron and does not have deep characteristics.

The study of activation function is an active field. Sigmoid and Tanh were the first activation functions. Later, Relu solved the deep learning problem of vanishing gradient. Recently, various activation functions have been developed, such as Leaky-relu [19], Selu [9], Elu [3], and Swish [26]. Relu is an excellent default choice, and the activation function of the network proposed here is also Relu. Several common activation functions and their derivatives are shown in Fig. 6.

Since both Sigmoid and Tanh have exponential calculations, the calculation speed is slow. Additionally, the gradient is less than one, so the gradient tends to vanish as it propagates through layers. Relu is a piecewise function, which means there is a linear function for each segment, but the overall Relu function is nonlinear, and the forward propagation and backward feedback are fast. The gradient equals one where the value is greater than zero, thus, there are no such problems as vanishing gradient or gradient explosion.

3.5 Dropout

Bagging is a technique for reducing generalization errors by combining multiple models. The main idea is to train several different models separately and then use all the models to vote for the output of the sample.

In general, combining multiple models is more effective for the following reasons:

1. Statistically, the hypothesis space of a learning task is large, and multiple hypotheses have the same performance on the training set. The use of a single model may result in poor generalization performance due to misclassification.

Fig. 6 Several common activation functions (blue represents the original function, and orange represents the corresponding derivative)

2. A single model may fall into local optimum, but integration can reduce risk.
3. The real hypothesis may not be included in the learning algorithm hypothesis. By integrating and expanding the learning into the hypothesis space, better approximation can be achieved.

As a Bagging technique, Dropout [7] randomly invalidates some of the neurons during the training process. That is, without parameter calculation, neurons that fail in training in each step are random. As shown in Fig. 7, after the $\times 1$ and $\times 2$ neurons in hidden layer 1 are discarded, hidden layer 2 does not receive input from these two neurons, and the next step may discard other neurons.

Dropout can be thought of as training multiple smaller and simpler models, and predicting is equivalent to combining multiple simple models.

3.6 Loss function

The role of the Softmax layer is to map the output of each class, which is a probability between zero and one; the sum of all class outputs is equal to one. The expression is as follows:

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^K e^{z_k}} \quad (4)$$

In classification, the last layer generally uses Softmax to obtain the probability of each class, and the loss function is the cross entropy. The cross entropy reflects the similarity of the two distributions. When there are two distributions, their cross entropy is

$$H(p, q) = - \sum p \ln q \quad (5)$$

Fig. 7 Dropout schematic

4 Results and discussion

4.1 Experimental environment

The software environment of this experiment is an Ubuntu16.04 operating system. The CPU processor is Intel Core i7-3770 8, and the memory is 12 GB. The graphics card is a NVIDIA GeForce GTX 1080, and the memory is 8 GB. The experiment is based on the open source deep learning framework Tensorflow, version 1.5. The CUDA version is 9.0, and the cuDNN version is 7.0.

4.2 Experimental data set

In order to realize the use of neural networks for fog identification, it is necessary to collect a large number of fog-containing and fog-free images. If the size of the data set is too small, over-fitting is likely to occur. Therefore, in order to meet the training requirements, a large number of images of natural scenes were collected from ImageNet. Some samples are shown in Fig. 8. ImageNet is an image database organized according to the WordNet hierarchy (currently only nouns), where each node of the hierarchy is represented by hundreds and thousands of images, and each node currently has an average of more than 500 images.

In order to simulate a fog-containing image, it is necessary to obtain an imaging model of agglomerate fog. In fact, the information acquired by the sensor includes two aspects. One is the information transmitted by the particle from the original optical path, and the other is the influence of atmospheric light, as shown in Eq. (6) [6]:

$$I(x) = J(x)t(x) + A(1 - t(x)) \quad (6)$$

where I represents a fog-containing image, J is the information reflected by objectives, i.e., a fog-free image, A is the atmospheric light, and t represents the attenuation effect of fog, i.e., the transmittance.

Fig. 8 Examples of natural scene images

Fig. 9 Generated fog-containing image

Fig. 10 Examples of the test data

Using Eq. (6), a fog-containing image can be generated using a fog-free image. Assume that I and J are in the range of $[0, 1][0, 1]$. Let the atmospheric light be $A = 1.0$ when generating data. Therefore, the random global transmittance is $t \in [0, \text{thres}]$, and thres is the upper limit of t . Figure 9 shows the corresponding fog-containing image when $\text{thres} = 0.7$.

Two sets of data were collected. The first data set (hereinafter referred to as test data set 1) was extracted from the above data, and the distribution was consistent; the second data set (hereinafter referred to as test data set 2) used the data set in [33]. The data in test data set 2, which was obtained from the actual outdoor scene, contained 263 fog-containing images and 289 fog-free images. Some fog-containing images in test data set 2 are shown in Fig. 10.

4.3 Training methods

A corresponding fog-containing image was generated from the input fog-free image. The size of a batch was 64, which consisted of 32 fog-free images and 32 corresponding fog-containing images. All images were scaled to a size of 256*256.

We used the Adam optimizer [8]; the learning rate was set to 0.0001, and the trained epoch was 50. Adam (adaptive moments) is a learning rate adaptive algorithm. Momentum directly incorporates the gradient first-order moment (exponential weighting). It also includes offset correction, which corrects the first-order moment (momentum term) initialized from the origin and the second-order moment estimate (non-center). Adam has strong robustness to hyper parameters and only needs modification in learning rates.

The dropout layer was added to the last two layers of fully connected layers, and the dropout rate was set to 0.5.

4.4 Results and discussion

To assess the performance of the proposed method, the indexes including precision (P) and recall (R) are employed here. They are defined as follows:

$$P = TP / (TP + FP)$$

$$R = TP / (TP + FN)$$

TP: true Positive, that is labeled as Positive and predicted as Positive.

TN: true Negative, that is labeled as Negative and predicted as Negative.

FP: false Positive, that is labeled as Negative and predicted as Positive.

FN: false Negative, that is labeled as Positive and predicted as Negative.

Experiments were performed according to the parameter settings in Sect. 4.3. The accuracy of the training set and the accuracy of the validation set are shown in Fig. 11. The curve in the graph is slightly fluctuating, which is normal because a batch of data was used for testing.

Fig. 11 Test results (the abscissa is the number of training steps, and the ordinate is the accuracy)

Table 1 Detection results with different blocks

Number of the blocks	Test data 1 (P,R)	Test data 2 (P,R)
4	(94.3%, 90.3%)	(87.5%,85.3%)
9	(97.6%,94.4%)	(91.25%, 89.5%)
16	(96.9%,93.2%)	(91.5%,89.4%)

Fig. 12 Some test images for blocking strategy**Table 2** Detection results with different blocks

Number of the blocks	P	R
1	90.1%	82.3%
9	97.7%	86.3%

Table 3 Comparison of test results with or without dropout (data 1)

	No fog (P,R)	Contains fog (P,R)	Overall (P,R)
Using dropout	(98.4%,95.8%)	(96.8%,93%)	(97.6%, 94.4%)
without dropout	(96.2%,93.9%)	(96.0%,93.7%)	(96.1%,93.8%)

For the test of the number of blocks, we divide the image into 4, 9 and 16 sub-images evenly, and use data 1 and data 2 to test the overall detection accuracy of the foggy image. The statistical results are shown in Table 1.

As shown in Table 1, when the number of image blocks is 4, the detection performance is the lowest. For data 1, when the number of image blocks is 9, the detection capability is the highest. For data 2, the detection capability is the highest when the number of blocks is 16, but there is only slight difference between the two when the number of blocks is 9. Therefore, the number of sub-images is set to 9 in this paper.

To verify the effectiveness of image blocking, we have added some additional test images (40 in total). Only part of these images are covered by fog. Some of these test images are shown in Fig. 12. Two strategies of block (9 blocks) and no block are adopted for retest. The experimental results are shown in Table 2.

In Table 2, the number “1” means that the entire image as input. As can be seen from Table 2, the effect of using the blocking strategy is significantly improved in both precision and recall compared with the case without blocking, which verifies the effectiveness of the image blocking processing proposed in this paper.

Using dropout should allow the model to avoid overfitting and achieve better performance. A comparison of performance is shown in Table 3 (data 1 was used to test). As we can see from Table 3, the dropout operation did achieve higher accuracy.

For simple tasks, a simple network is often used to obtain higher accuracy. If a complex network is used, it may cause over-fitting and the gradient may not be returned. To test this, the network layer was deepened in this paper, and the network structure comparison is shown in Table 4. The experimental parameters of the two networks were the same. Net1 achieved a precision of 97.6% and recall 94.4% as mentioned earlier, and Net2 only achieved a precision of 96.1% and recall 93.8%. These results demonstrated that it is not suitable for very deep networks to classify a two-category task.

For the real-time data set (test data set 2), the test results are shown in Table 5. Due to the inconsistency between the training data and the test data, the test result was lower than that of the test data set 1. However, for both fog-containing and fog-free images, the precision was greater than 90%.

To further verify the effectiveness of the proposed method, several representative algorithms [13, 37, 38] mentioned in the introduction were selected as the comparison methods. The performance of different methods tested on data 1 and test data 2 is shown in Table 6.

Table 4 Comparison of network structures

Original network Net1	Deepened network Net2
Conv 7*7, 96	
MaxPooling	
Conv 5*5, 256	Conv 5*5, 256
	Conv 5*5, 256
MaxPooling	
Conv 3*3, 384	Conv 3*3, 384
	Conv 3*3, 384
MaxPooling	
	Conv 3*3, 512
	Conv 3*3, 512
	MaxPooling
Full Connect, 512	
Full Connect, 512	
Full Connect, 2	

Table 5 Test results of test data set 2

	No fog	Contains fog	Overall
Precision	92.3%	90.2%	91.25%
Recall	90.2%	88.8%	89.5%

Table 6 Comparison of different methods

Method	Test data 1	Test data 2
CNN-RNN [38]	(94.1%, 93.3%)	(89.2%, 88.4%)
RSCM [13]	(93.6%, 92.0%)	(88.5%, 88.8%)
GSLSNet [37]	(95.3%, 95.1%)	(90.4%, 89.9%)
Proposed method	(97.6%, 94.4%)	(91.25%, 89.5%)

The methods [13, 37, 38] output many weather types including fog. For comparison, the detection accuracy of fog image is only given here. From Table 6, it can be seen that the above four methods can achieve high detection accuracy. Both for data 1 or data 2, the detection accuracy of the proposed method has obvious advantages compared with the others. At the same time, it can be found that the test performance of data 1 is higher than that of data 2, no matter the proposed method or other methods. This shows that the actual data is more complex than the simulation data.

5 Conclusion

The image detection algorithm of fog hazard USES the roadside video surveillance to realize the prediction and early warning. The algorithm has the characteristics of fast and accurate emergency response. With the characteristics of fast and accurate emergency response, this algorithm can play a role in disaster prevention and mitigation, and provide safety services for vehicle travel, which has important practical application value. An agglomerate fog detection method based on the convolutional neural network were described in this paper. The work includes an analysis of each network layer, selection and analysis of the activation function, and selection and analysis of the loss function. After an image was divided into sub-areas (blocks), the proposed network is used to fog detection on each block, and then combined the determination results of all sub-areas to obtain a final result. The test results of simulation data and real data show that the image detection method proposed in this paper has high detection accuracy.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Bay H, Tinne T, Luc VG (2006) Surf: speeded up robust features. In: European Conference on Computer Vision. Springer, Berlin
2. Cai B, Xu X, Jia K, Qing C, Tao DD (2016) An end-to-end system for single image haze removal. *IEEE Trans Image Process* 25(11):5187–5198
3. Clevert D A, Thomas U, Sepp H (2015) Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv.2015, 1511.07289*
4. Dalal N, Bill T (2005) Histograms of oriented gradients for human detection. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2005. Vol 1*. IEEE
5. He K (2016) Deep residual learning for image recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition*.
6. He K, Sun J, Tang X (2009) Single image haze removal using dark channel prior. In: *IEEE Conference on Computer Vision and Pattern Recognition, Miami, USA, pp 1956–1963*
7. Hinton GE (2012) Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv. 1207.0580*

8. Kingma D P, Jimmy B (2014) Adam: a method for stochastic optimization. arXiv preprint arXiv.2014, 1412. 6980.
9. Klambauer G, Unterthiner T, Mayr A (2017) Self-normalizing neural networks. In: Advances in neural information processing systems, pp 972–981
10. Kwang Y C, Kyeong M J, Byung C S (2017) Fog detection for de-fogging of road driving images. In: IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)
11. LeCun Y (1998) Gradient-based learning applied to document recognition. Proc IEEE 86(11):2278–2324
12. Li, B (2017) An All-in-One Network for Dehazing and Beyond. arXiv preprint arXiv. 1707.06543
13. Lin D, Lu C, Huang H, Jia J (2017) RSCM: region selection and concurrency model for multi-class weather recognition. IEEE Trans Image Process 26(9):4154–4167
14. Lu X, Ma C, Ni B, Yang X (2019) Adaptive region proposal with channel regularization for robust object tracking. IEEE Trans Circuits Syst Video Technol. <https://doi.org/10.1109/TCSVT.2019.2944654>
15. Lu X, Wang W, Ma C, Shen J, Shao L, Porikli F (2019) See more, know more: unsupervised video object segmentation with co-attention siamese networks. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 2019, pp 3618–3627. <https://doi.org/10.1109/CVPR.2019.00374>
16. Lu X, Ma C, Shen J, Yang X, Reid I, Yang MH (2020) Deep object tracking with shrinkage loss. IEEE Trans Pattern Anal Mach Intell. <https://doi.org/10.1109/TPAMI.2020.3041332>
17. Lu X, Wang W, J. Shen, Tai YW, Crandall DJ, Hoi SCH(2020) Learning video object segmentation from unlabeled videos. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, pp 8957–8967. <https://doi.org/10.1109/CVPR42600.2020.00898>.
18. Lu X, Wang W, Danelljan M, Zhou T, Shen J, Van Gool L (2020) Video object segmentation with episodic graph memory networks. In: Vedaldi A, Bischof H, Brox T, Frahm JM (eds) Computer vision—ECCV 2020. ECCV 2020. Lecture Notes in Computer Science, vol 12348. Springer, Cham. https://doi.org/10.1007/978-3-030-58580-8_39
19. Maas A L, Hannun A Y, Ng A Y (2013) Rectifier nonlinearities improve neural network acoustic model. In: 2013 Proceedings of ICML, vol 30(1), p 3
20. Mario P, Heidrun B, Gerhard R, Slobodan I (2012) Image based fog detection in vehicles. In: 2012 Intelligent Vehicles Symposium Alcalá de Henares, Spain, June 3–7. pp 1132–1137
21. Ojala T, Matti P, Topi M (2017) Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Trans Pattern Anal Mach Intell 24(7):971–987
22. Parkhi OM, Andrea V, Andrew Z (2015) Deep face recognition. BMVC 1(3)
23. Rakesh A, Ramesh KS, Lakhan DS, Aman K (2016) Fog detection using GLCM based Features and SVM. In: 2016 Conference on Advances in Signal Processing (CASP) Cummins College of Engineering for Women, Pune, Jun 9–11, 2016. pp 72–76
24. Ren W Q (2016) Single image dehazing via multi-scale convolutional neural networks. In: European Conference on Computer Vision. Springer, Cham
25. Salma A, Abdelhak E, Fouad E (2016) Local fog detection based on saturation and RGB-correlation. In: 2016 13th International Conference Computer Graphics, Imaging and Visualization, pp 1–5
26. Ramachandran P, Barret Z, Quoc V L (2018) Searching for activation functions. arXiv preprint arXiv.2018. 1710.05941
27. Rita S, Carsten K, Su BP, Jason JY (2014) Fast fog detection for camera based advanced driver assistance systems. In: 2014 IEEE 17th International Conference on Intelligent Transportation Systems (ITSC) October 8–11. Qingdao, China. pp 1369–1374
28. Simonyan K, Andrew Z (2014) Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv. 1409.1556
29. Szegedy C (2015) Going deeper with convolutions. Cvpr, 2015
30. Szegedy C (2016) Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition
31. Szegedy C (2017) Inception-v4, inception-resnet and the impact of residual connections on learning. AAAI, vol 4
32. Szegedy C (2017) Inception-v4, inception-resnet and the impact of residual connections on learning. AAAI 4:2017
33. Tian Y, Xia D, Xu Y (2014) Single foggy image restoration based on spatial correlation analysis of dark channel prior. J Syst Eng Electron 25(04):688–696
34. Xie S (2017) Aggregated residual transformations for deep neural networks. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR)

35. Yang B, Zhang S, Tian Y et al (2019) Front-vehicle detection in video images based on temporal and spatial characteristics. *Sensors* 19(7):1728–
36. Yang B, Tang M, Chen S et al (2020) A vehicle tracking algorithm combining detector and tracker. *EURASIP J Image Video Process* 2020(1):1–20
37. Yu T, Kuang Q, Hu J, Zheng J, Li X (2020) Global-similarity local-saliency network for traffic weather recognition. *IEEE Access*. <https://doi.org/10.1109/ACCESS.2020.3048116>
38. Zhao B, Li X, Lu X, Wang Z (2018) A CNN-RNN architecture for multi-label weather recognition. *Neurocomputing* 322:47–57

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.