


RESEARCH

Open Access



Visibility estimation via deep label distribution learning in cloud environment

Mofei Song^{1,2*} , Xu Han^{1,2}, Xiao Fan Liu³ and Qian Li⁴

Abstract

The visibility estimation of the environment has great research and application value in the fields of production. To estimate the visibility, we can utilize the camera to obtain some images as evidence. However, the camera only solves the image acquisition problem, and the analysis of image visibility requires strong computational power. To realize effective and efficient visibility estimation, we employ the cloud computing technique to realize high-through image analysis. Our method combines cloud computing and image-based visibility estimation into a powerful and efficient monitoring framework. To train an accurate model for visibility estimation, it is important to obtain the precise ground truth for every image. However, the ground-truth visibility is difficult to be labeled due to its high ambiguity. To solve this problem, we associate a label distribution to each image. The label distribution contains all the possible visibilities with their probabilities. To learn from such annotation, we employ a CNN-RNN model for visibility-aware feature extraction and a conditional probability neural network for distribution prediction. The estimation result can be improved by fusing the predicting results of multiple images from different views. Our experiment shows that labeling the image with visibility distribution can boost the learning performance, and our method can obtain the visibility from the image efficiently.

Keywords: Cloud computing, Deep learning, Label distribution learning, Visibility estimation

Introduction

Meteorological visibility is a crucial index for reporting daily air quality, which has an important bearing on environmental protection. Visibility has a wide range of applications such as traffic safety [1], industrial, agricultural production, and smart city. Traditionally, visibility can be estimated by some specialized equipment, such as a transilluminator or foreword scattering sensor [2]. However, since the equipment is usually expensive and inconvenient, we can place the equipment at only a few weather stations to detect the visibility of some fixed scenes. This cannot satisfy the requirement of multiple monitoring applications. To realize ubiquitous and intelligent monitoring as done in [3], we can utilize abounding and budget

cameras as an alternative. By analyzing the image taken by these cameras, we can obtain direct evidence for visibility estimation effectively and efficiently.

The image-based visibility estimation constructs the mapping between the image representation and the visibility value. At present, many image-based visibility analysis methods are proposed [4–6], and they focus on creating some complex models to extract the visibility-aware cues from the image. The recent trend is geared towards using deep learning [7, 8] to obtain the visibility-aware feature. These methods assume the computing power is enough, and the actual deployment problem is not considered. The normal cameras cannot realize the image analysis processing due to their limited computational power. It is impractical to assign an external computer for every camera, since even powerful CPUs cannot handle deep learning tasks efficiently. Usually, a graphics processing unit (GPU) is required but its cost is much higher. Another way is to use some programmable cameras with

*Correspondence: songmf@seu.edu.cn

¹the School of Computer Science and Engineering, Southeast University, Nanjing, China

²the Key Lab of Computer Network and Information Integration (Ministry of Education), Southeast University, Nanjing, China

Full list of author information is available at the end of the article

the embedded computing device. However, a typical deep model is difficult to be deployed on such devices. Due to its limited resource, the model is required to be compressed according to its storage space [9, 10], which might reduce the performance. Meanwhile, the inference speed is slow, since its computational capacity is much worse than the external devices having dedicated graphics cards. Moreover, the deployed model is hard to be updated since the model is stored in the offline device. This brings a high cost to the operation and maintenance of the visibility monitoring system when the model is upgraded. Therefore, it demands an efficient and effective way to estimate the visibility of the image.

To realize the real-time image-based visibility estimation, an effective way is cloud computing. Cloud computing is a distributed computing paradigm for sharing configurable computing resources. By such a technique, data partitioning and sampling [11] are utilized to improve the processing of big data. At the same time, the images generated from multiple cameras can be analyzed with high parallelism and programmable flexibility through the distributed architecture [12] and fast data exchange [13]. Recently, cloud computing has been used in video surveillance for traffic services [14], which improves the response efficiency of the services. Inspired by these methods, our method integrates cloud computing into the image-based visibility monitoring process. This alleviates the lack of computing power in the visibility estimation applications.

However, it is difficult to deploy an effective visibility estimation model in the cloud environment. Currently, most visibility estimation methods use deep learning to construct the visibility prediction model [7, 8]. Since these methods formulate the visibility estimation as a regression problem, we should label the visibilities of all the training images accurately. However, it is difficult to obtain the precise ground truth of image visibility. The human specification of absolute visibility from a single image is unreliable [15]. The specialized equipment cannot generate accurate visibility labels either due to deployment variations and environmental influences. The annotation

problem brings many challenges to the construction of a cloud computing system for visibility estimation.

To overcome the problem, one way is to use image pairs labeled by ranking their visibilities as the supervision [16]. Though it is much easier to achieve the relative visibility annotation given a pair of images, absolute visibility cannot be derived directly from the ranking information. Besides, the annotation burden of relative visibility is increased significantly since there are lots of image pairs to be annotated.

Inspired by a novel machine learning paradigm called Label Distribution Learning (LDL) [17], we propose to label an image by a mixture of visibility values with different intensities, which is described as a distribution. Since such label distribution contains several possible visibility values, the problem of inaccurate annotation is overcome. To obtain such a label, we transform an absolute visibility label provided by humans or equipment into a visibility distribution. The transformation is based on the following observation: the images with close visibility have a high degree of similarity. Accordingly, we adopt one-dimensional Gauss distribution for visibility annotation, where the absolute visibility label is the mean of the distribution (as shown in Fig. 1). Thus, the absolute visibility has the highest intensity, while the relevant degree of other visibilities is inversely proportional to their distances to the absolute label.

Compared with previous labeling types, the label distribution has two superiorities for visibility estimation. On the one hand, the label distribution considers both the ambiguity of the visibility label and the convenience of the visibility prediction. The label distribution improves the robustness of model learning with the uncertain label. Meanwhile, absolute visibility can be obtained directly by searching the highest intensity from distribution. On the other hand, the label distribution can provide more informative supervision. Since every training image labeled by particular visibility is relative to the other closer visibility, it can affect the learned model of adjacent visibilities. In other words, the training images for particular visibility

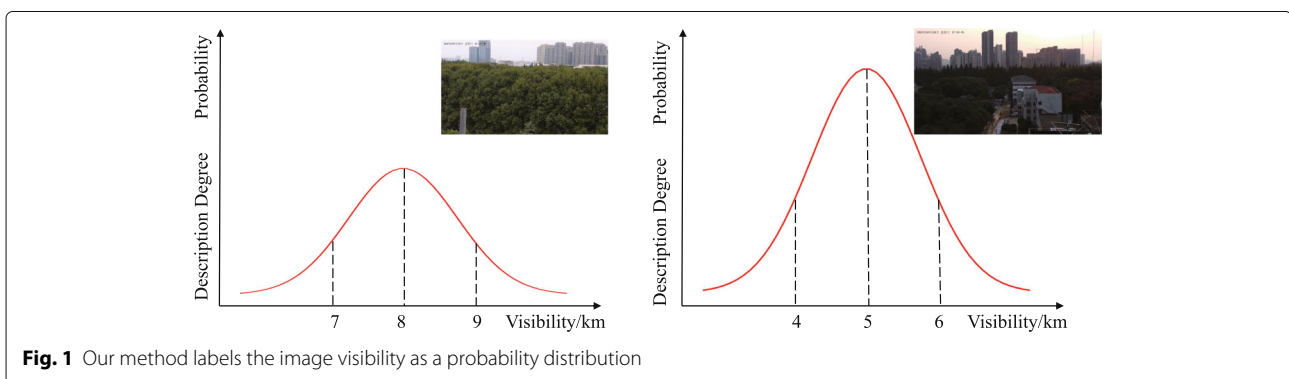


Fig. 1 Our method labels the image visibility as a probability distribution

are substantially increased without extra image collection and annotation.

In this paper, a novel visibility estimation method using deep LDL in a cloud environment is proposed. The method employs cloud computing to process the visibility image data, which satisfies the real-time requirements of meteorological monitoring applications. To obtain the visibility estimation model for the clouding center, we use deep label distribution to train the deep neural network. Given a set of images labeled by absolute visibility, we first transform the absolute label into a label distribution. Then, we integrate LDL, CNN, and RNN into a unified framework for visibility model learning. The LDL module focuses on the label level and employs the label distribution to remove the label ambiguity. The CNN module focuses on the global feature level and is responsible for learning the overall visibility from the whole image. The RNN module focuses on the local region level and searches the farthest region to give richer information for visibility estimation. By integrating the three levels of information including the label, global feature, and local region, we can construct a more effective image representation to estimate the visibility. The learned model can generate the distribution of all the possible visibilities given a test image. We finally use the visibility with the highest probability as the predicted visibility. To improve the robustness of the automatic estimation, the Dempster-Shafer theory is used to fuse the visibility estimation results of multiple images obtained from different views at the same location and time.

The contributions of this study are summarized as follows: 1) We propose to adopt cloud computing to deal with image-based visibility estimation, which improves real-time performance. 2) We propose to use the label distribution as the supervision for visibility estimation, which can not only overcome the inaccurate annotation problem, but also boost the learning performance without the increase of training examples. 3) We utilize the Dempster-Shafer theory to fuse multiple predictions of the images from different views, which promotes the stability and robustness of the algorithm.

Related work

This work relates to several research areas. In this section, we briefly review the existing work on image-based visibility estimation, cloud computing and evidence theory.

Image-based visibility estimation

The early image-based visibility estimation methods use some hand-crafted features to achieve visibility, such as contrast, image inflection point, or dark channel. For example, Busch et al. [18] employ wavelet transform to extract the contrast information of the image edge for visibility estimation. Jourlin et al. [19] select some critical

points to construct a scene depth map with the stereo vision method. Bronte et al. [20] measure the visibility in poor weather conditions by calculating the vanishing point of the horizon as the farthest visible pixel. Graves et al. [21] train a log-linear model by combining local contrast features and dark channel prior. Xiang et al. [22] integrate average Sobel gradient operator and dark channel theory to detect the daytime visibility level. These methods focus on low-level image processing techniques, and cannot lead to the practical and stable estimation result across different scenes.

To improve the performance of visibility estimation, some researchers create some physic probabilistic models by meteorological law. Babari et al. [4, 5] employ a non-linear regression model called Koschmieder's theory, which describes the relationship between the contrast distribution and visibility. The measurement of the image contrast can be improved further by the extinction coefficient [6]. However, the performance of these physics models is affected by weather conditions, illumination, and scene variations. It is extremely difficult to hard-code the enormous variability of these complex factors in general.

To improve the adaptiveness, a more sophisticated way is to extract the visibility prediction model from data by deep learning. Li et al. [7] employ a generalized regression neural network (GRNN) and a pre-trained CNN model for visibility estimation. Giyenko et al. [23] detect the visibility range of the weather images by building Shallow CNN neural network with fewer layers. Palvanov et al. [8] use three streams of deep integrated convolutional neural networks to extract the visibility-aware feature from spectrally filtered images, FFT-filtered images, and RGB images. Wang et al. [24] use a multimodal CNN architecture to learn the visibility-aware features by combining the two sensor modalities. Li et al. [25] propose a transfer learning method based on the feature fusion for visibility estimation. Lo et al. [26] further introduce PSO feature selection into the transfer learning method to improve the performance. These methods all treat the visibility estimation problem as a regression model, which is suffered from the label ambiguity challenges. To solve it, You et al. [16] propose a relative CNN-RNN model to learn relative atmospheric visibility from images. They ask the annotators to rank two given images based on the visibility by the crowdsourcing technique [27]. Though the label ambiguity is eliminated by such supervision, the annotation cost is increased significantly. Our method is inspired by these approaches, but we propose to use label distribution as the supervision. This provides a cheap and effective way to resolve the label ambiguity issue in visibility estimation.

Moreover, meteorological visibility is also temporal correlated, which can be predicted by some time-aware

dynamic analysis techniques [28, 29]. For example, Wu et al. [30] utilize the environmental state information to achieve the visibility prediction results in the airport. In contrast, our method does not rely on temporal information and estimates the visibility from the real-time images.

Cloud computing

Cloud computing is a new computing paradigm, which can process billions of data in seconds and provide strong network service [1, 14]. It can effectively reduce offloading and maximize the system utility by balancing resource allocation [31, 32]. With the rapid development of cloud computing, more user records are generated and stored in the data center for other applications [33]. Cloud computing has been widely used in many applications with high load and high concurrencies, such as smart city [34], intelligent manufacturing [35], and internet of thing [36]. Based on its advantages for large-scale data processing, we integrate cloud computing and image processing into a whole visibility monitoring framework with a huge network of cameras, which improves the real-time and accurate performance of visibility estimation.

Information fusion

Since a single sensor system is unstable, it is difficult to meet the demands of the complex environment. To improve the robustness of the system, information fusion is used to combine the prediction results from multiple sensor devices [37]. Such a technique can make multiple sensors cooperate with each other and produce more accurate results by synthesizing the complementary information [38, 39]. There are many information fusion techniques, such as Dempster-Shafer theory [40], fuzzy logic [41] and rough set theory [42]. Among these techniques, we choose Dempster-Shafer (D-S) theory for its high fusion accuracy and flexibility. By the confidence function and the likelihood function, it can calculate the uncertainty interval to describe the trust distribution of different information. Currently, D-S theory has been widely used in event probability prediction [43], image target recognition [44] and case decision analysis [45]. Our method uses D-S theory to combine the predictions of multiple images from different views. When the visibility information derived from different images is inconsistent, D-S theory can extract the homogenous information through common credibility. This ensures that the accuracy of the fused result is better than the original single result. To the best of our knowledge, this is the first time to introduce the D-S theory into image-based visibility estimation.

Method

Overview

We propose a visibility estimation method based on deep LDL in a cloud environment. The overall idea is that we combine cloud computing and image process to estimate visibility efficiently. With cloud computing, we can make the front-end monitoring device thinner. The camera is only responsible for capturing and compressing the images. Then, the high-performance network is utilized to transmit the image data from the camera to the cloud platform. Thus, the complex image analysis task can be performed quickly and remotely in the cloud with high performance computing cluster. The results can be feedbacked to the monitoring center. This alleviates the lack of front-end computing power in the visibility estimation application. Our developed method saves the required resources by utilizing high-performance parallel computing of graphics processing units (GPU) in cloud computing services. With the GPU cluster, we can adopt larger and deeper models to improve the accuracy of visibility estimation without sacrificing efficiency. Meanwhile, another advantage of cloud computing is the convenience of system maintenance. Compared with front-end computing in the local PC or programmable cameras, the deep model is much easier to be upgraded, which helps to ensure the stability and consistency of the system.

Figure 2 illustrates the whole pipeline of our method. The camera nodes collect the images and transmit them to the cloud data center. These cameras can adjust their angles continuously to obtain different images. Since these images are generated at the same location and time, they have nearly similar visibility. These images are analyzed by the pre-trained deep model stored in the data center. The visibility at the location of the camera node is finally achieved by fusing all the predictions of multiple images from different views.

To train the deep model, we label the image visibility with a distribution vector, and integrate deep learning and LDL for visibility estimation. The training input is a training image set S , and every image $x \in S$ is labeled by an absolute visibility y . We also annotate the farthest region of every image by the coordinates of the bounding box b . The training output is a visibility estimation model, which includes a deep CNN-RNN module and an LDL module. Overall, the training session contains three parts: label transformation, feature learning, and distribution learning. In the following, we first describe each part of the training stage in detail, and then introduce the prediction fusion method when estimating the visibility.

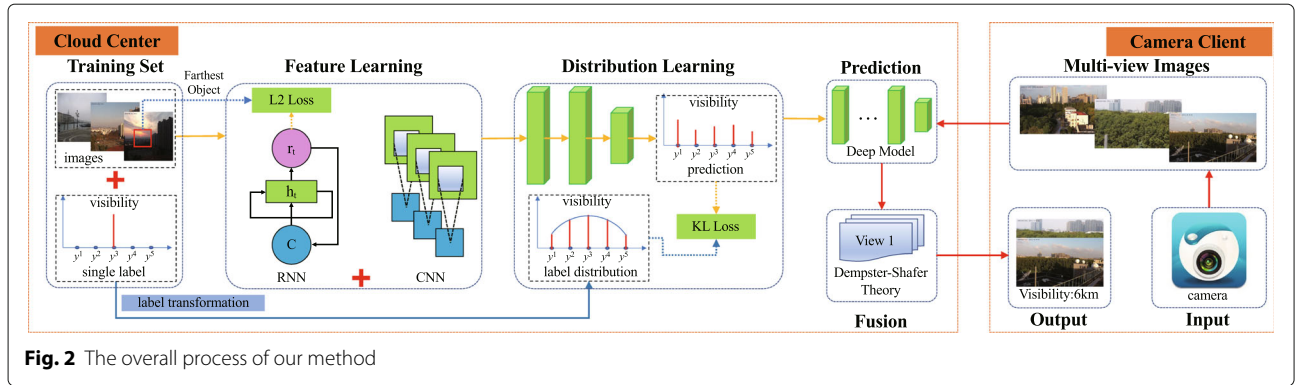


Fig. 2 The overall process of our method

Label transformation

The goal of label transformation is to generate a label distribution for every training image x . Given an image x , let the vector $D = \{d_x^{y_1}, d_x^{y_2}, \dots, d_x^{y_c}\}$ denote its label distribution, where, $Y = \{y_1, y_2, \dots, y_c\}$ is the label space, c is the size of the label space, $d_x^{y_i} \in [0, 1]$ is called description degree of the image x , which means the probability of its visibility being y_i . Due to the complexity of the label distribution, fully manual annotation is impractical. Thus, we prefer an automatic way to generate the distribution from an absolute visibility label y .

For visibility estimation, the label space is intrinsically continuous. To ease the learning, we quantize the continuous label space into a discrete space Y with equal step size Δy . In our method, we set the label space $Y = [1km : \Delta y : 12km]$ (MATLAB notation) and the step size $\Delta y = 0.1km$. Thus, the label distribution D of every image x is a 111D vector, which satisfies $\sum_i d_x^{y_i} = 1$.

To generate the label distribution D , we first determine the distribution type according to the characteristic of the visibility estimation problem. Given an image x labeled by an absolute visibility y , it is reasonable to make the corresponding description degree $d_x^y \in D$ highest in the final distribution D . Meanwhile, since the neighbor visibility looks similar in appearance, we can increase the description degrees which are adjacent to the label y . Naturally, the description degree $d_x^{y_i}$ should be gradually reduced when the visibility y_i is far away from the label y . Based on this observation, we choose the probability density function of one-dimensional Gauss distribution for label transformation.

The one-dimensional Gauss distribution is a type of continuous probability distribution for a real-valued random variable. Its general form is shaped as:

$$f(y_i) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y_i - \bar{y})^2}{2\sigma^2}\right) \quad (1)$$

where, the parameter \bar{y} is the mean of the distribution, and σ is its standard deviation.

According to Eq. 1, we transform the absolute label y into a probability vector. Since we expect the description degree $d_x^y \in D$ is the highest, we set the parameter \bar{y} as the absolute label y . The parameter σ is a hyper-parameter, which is optimized by a simple grid searching described in the experiment section. After determining these two parameters, we can compute the probability density when the random variable equals y_i as its description degree $d_x^{y_i}$. In other words, we compute the description degree $d_x^{y_i}$ by

$$d_x^{y_i} = f(y_i) \quad (2)$$

Such label distribution D is a discrete distribution by a dense sampling of one-dimensional Gauss distribution. The discretion process makes the sum of the vector D not equal to 1, which achieves an illegal probability density function. To force $\sum_i d_x^{y_i} = 1$, this vector is numerically normalized to generate the expected label distribution D . The distribution D is used for the following learning process, which makes the distribution D' derived from the deep network consistent with the distribution D .

Feature learning

To simulate the procedure of humans judging the visibility, we follow the relative CNN-RNN method [16]. Since the method uses a ranked image pair to train the deep network, its network architecture contains two similar CNN-RNN branches. Instead, our method uses only one CNN-RNN model to extract the visibility-aware feature, which is more efficient.

Specially, the CNN-RNN model imitates the coarse-to-fine way to detect the farthest target from the image. The CNN module learns the overall visibility from the global image, while the RNN model simulates the region search to realize the coarse-to-fine attention shift, as shown in Fig. 3. There are also other advanced methods for the region searching, such as temporal CNN [46], LSTM [47] and GRU [48]. The temporal CNN needs to maintain the sequence into the memory, while LSTM and GRU contain more parameters than RNN. Thus, we choose

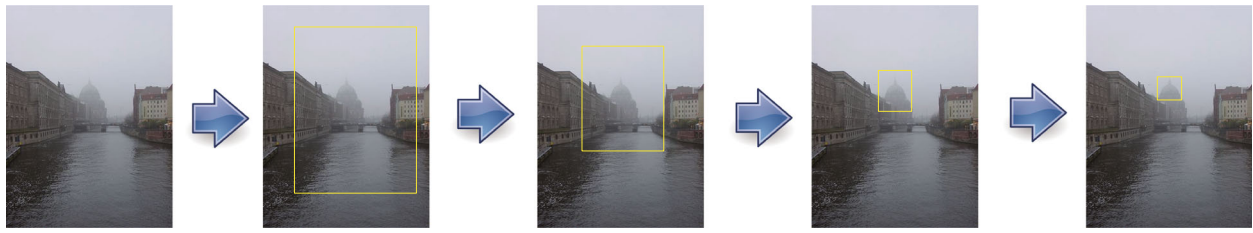


Fig. 3 The RNN module detects the farthest target from the image

RNN for its fewer parameters and higher efficiency. In the experiment, we find RNN to work well in practice. By combining the CNN and RNN model, the final global feature can be more sensitive to the farthest region in the image, which is an important cue for visibility estimation.

Figure 4 illustrates the architecture of our method. For the CNN module, we follow the design of AlexNet [49] for global feature extraction. Our CNN module contains 7 layers, which includes 5 convolution layers and 2 fully connected layers.

For the RNN module, we construct K layers for each of the first six CNN layers, and 1 layer for the last CNN layer. Totally, there are $6K + 1$ states in a sequential manner in the RNN module. Every state predicts a bounding box r_t , and the list of the bounding boxes shows the searching process of the farthest region, namely, from the whole image to the farthest region.

At each recursive step, the RNN module first crops the image x into a sub-region c_t^x based on the predicted bounding box r_{t-1} , which is the location result of the previous state. Then, the internal state h_t is updated by the core network g_h according to the sub-region c_t^x and the historical state h_{t-1} . The generated state h_t encodes the knowledge of searching the farthest region. Finally, we use the internal state h_t to predict the next bounding box r_t by the location network as $r_t = g_r(h_t)$, which is a two-layered network. To exchange the information between CNN and RNN, some shortcuts connections are added between the $(7 - i)$ th ($i = 0, \dots, 6$) layer of CNN and the $(Ki + 1)$ th state of RNN.

To train the RNN module, we need to indicate the ground-truth bounding box of every state. To this end, we assume that the list of the bounding boxes is evenly distributed. Accordingly, given the annotated bounding box b of the farthest region, we generate the whole ground-truth list of the bounding boxes $B = \{b_1, b_2, \dots, b_{6K+1}\}$ by average sampling. During the training phase, we expect to minimize the divergence between the predicted bounding boxes $\{r_1, r_2, \dots, r_{6K+1}\}$ and the ground truth. Accordingly, we define the objective function of RNN as a location L_2 norm loss:

$$L_l = \sum_{t=1}^{6K+1} \|b_t - r_t\|^2 \quad (3)$$

To integrate the information of CNN and RNN, the output of the CNN module and the last state h_{6K+1} of the RNN module are concatenated into a global vector representation f for visibility estimation.

Distribution learning

To utilize the label distribution D , we integrate it into the network architecture. Naturally, we can use several fully connected layers with a softmax layer to turn the feature f into a distribution directly. However, this yields lots of weights between the feature layer and the output layer, which makes it difficult to obtain the optimal solution. Thus, we follow the design of the conditional probability neural network (CPNN) [50].

As shown in Fig. 4, CPNN contains three fully connected layers. The input of CPNN contains the feature f and a discrete visibility label y . The output of CPNN is a single value $p(y|f)$ as the conditional probability. To realize the training, the Kullback-Leibler divergence is employed to measure the difference between the estimated distribution and the ground-truth distribution D :

$$L_d = \sum_j d_x^{y_j} \ln \frac{d_x^{y_j}}{p(y_j|f)} \quad (4)$$

Finally, we define the entire objective function as the sum of the location loss and the distribution loss:

$$L = L_l + L_d \quad (5)$$

Given the entire objective function, we simultaneously optimize the CNN-RNN module and the CPNN module through back-propagation with stochastic gradient descent [51]. To accelerate the training, we use the CNN pre-trained by ImageNet. According to the learned model, the predicted visibility could be the one with the maximum description degree.

Prediction fusion

Due to the diverse image quality and the inherent feature noise, the system cannot estimate the visibility from a

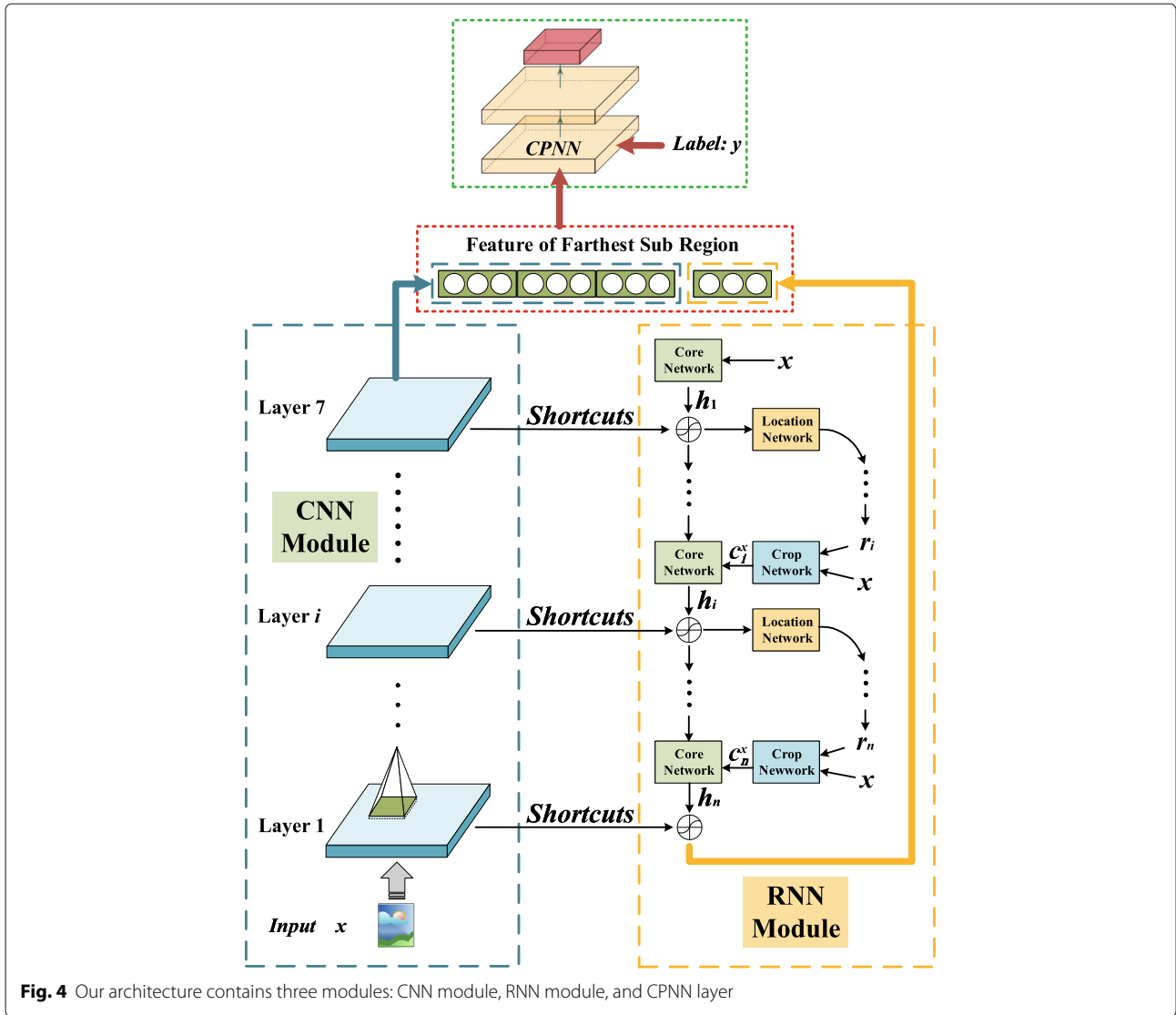


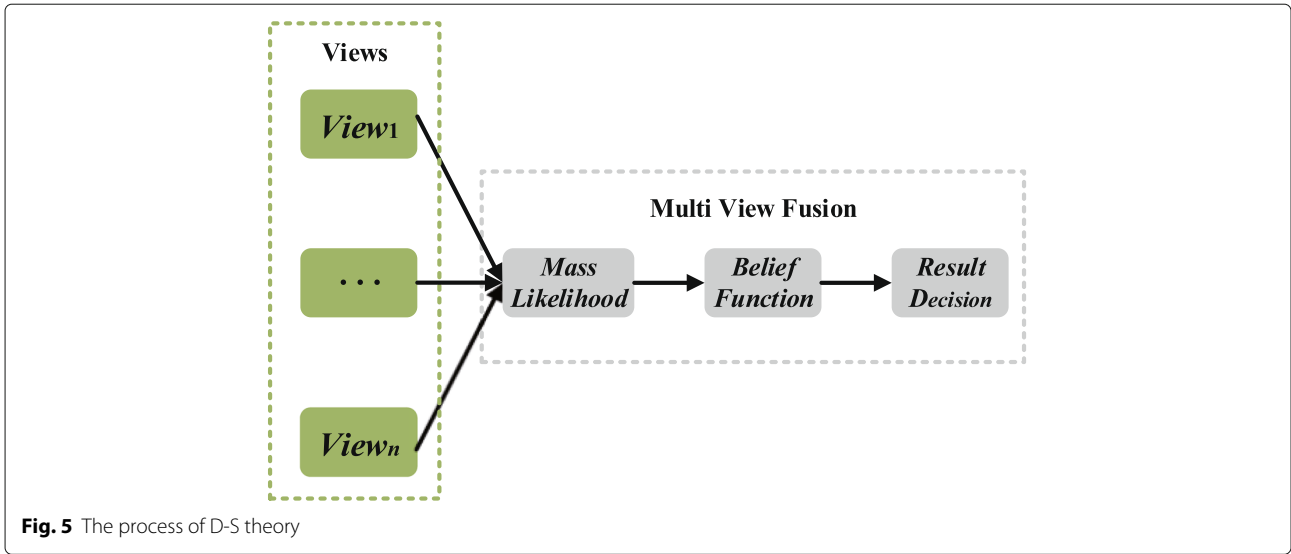
Fig. 4 Our architecture contains three modules: CNN module, RNN module, and CPNN layer

single image reliably. Currently, most cameras can adjust their internal parameters to obtain multiple images from different views simultaneously. Since these images are captured at the same location and time, they intrinsically have the same visibility. Based on this observation, we utilize the complementary information between multiple images from different views to enhance the accuracy of the visibility estimation.

To this end, we employ Dempster-Shafer theory to fuse multiple predictions derived from different images. There are two steps in Dempster-Shafer theory. First, a set of related propositions are created with their subjective probabilities as the degrees of belief. For visibility estimation, we can create a proposition for every image captured from different views. The proposition is shaped as: the visibility of the image is y . We can also take the output of CPNN as its degree of belief.

Second, the degrees of belief are combined based on mass likelihood and belief function, as shown in Fig. 5. The DS rule of combination corresponds to the normalized conjunction of mass functions. Based on Shafer's description, given two independent and distinct pieces of evidence on the same frame of discernment, their combination is the conjunctive consensus between them. When using the DS rule for multi-view visibility estimation, we can obtain the belief function of every proposition derived from every image. The belief function considers both the relevance and conflict between different views, which makes the fusion result more reliable.

The core of D-S theory is the construction of the proposition set. To realize it for multi-view prediction fusion, a direct way is to create c mutually exclusive propositions for every single visibility, where c is the size of



the label space Y . However, we found such propositions cannot always improve the performance, since the quality of prediction fusion is affected by the possible strong conflict between different views [52].

To solve it, we design several fuzzy propositions to reduce the evidence conflict. The fuzzy proposition is corresponding to a subset of the visibility label space, instead of single visibility. To avoid combinatorial explosion, we utilize the continuity of the label space to remove some subsets with low confidence. First, the subset should be a continuous sequence of visibility labels. Second, the length of the sequence should be equal to 3. The limitation of the length is selected empirically. If the parameter is too long, the fuzzy proposition contains very little information. Thus, we set it as 3. Accordingly, our proposition set A contains two types of propositions. The first type of proposition is unambiguous, which is indicated by A_j . The proposition A_j means that the visibility of multi-view images is $y_j \in Y$. The second type is fuzzy, i.e., the visibility of the image is one element of a continuous sequence. The fuzzy proposition is denoted by A_{jk} ($j < k \leq j + 2$), which means that the corresponding sequence is $\{y_j, \dots, y_k\}$. Obviously, we have $A_i \subset A_{jk}$ if $j \leq i \leq k$. To simplify the notation, we denote A_{jk} as A_j in the following session.

To support the fusion of such a proposition set, we first modify the input label of our CPNN to obtain the confidences of all the propositions. The original network can only take every single visibility y_j as the input. We expand the input label set by adding the continuous sequence that appears in the fuzzy proposition. To realize the training of the CPNN, we measure the ground-truth description degree $d_x^{y_{ij}}$ by the sum of all the related probabilities:

$$d_x^{y_{jk}} = \sum_{i=j}^k d_x^{y_i} \tag{6}$$

The deep network is re-trained by the extending input label set. Based on the deep network, we can obtain the normalized confidences of all the propositions given an image.

Then, we use Dempster’s rule to combine the pieces of evidence derived from multiple images. Given the multi-view image set $V = \{View_1, \dots, View_n\}$, our deep network generates the probabilities of all the propositions for each image $View_i$. We denote $m_i(A_j)$ as the confidence of the image $View_i$ supporting the proposition A_j . Thus, we can use $m_i(A_j)$ to define the mass function, which is the basic probability assignment function. Based on the mass function, we can compute the belief function $Bel(A_j)$. The belief function describes the confidence of all the images supporting the visibility estimation proposition, which is computed by the following rule:

$$Bel(A_j) = \frac{1}{1 - K} \sum_{\bigcap_i^n A_{k_i} = A_j} \prod_i^n m_i(A_{k_i}) \tag{7}$$

where, K is the normalization constant as:

$$K = \sum_{\bigcap_i^n A_{k_i} \neq \emptyset} \prod_i^n m_i(A_{k_i}) \tag{8}$$

Finally, we choose the proposition with the highest belief as the result of prediction fusion. If this proposition A_{jk} is fuzzy, we then choose its single subset A_i ($j \leq i \leq k$) with the highest plausibility as the fusion result.



Fig. 6 Some images in RMID. Every line of images are captured at the same location and time with different views

Experiments

Implementation detail

For the CNN part, it takes a long time to converge if random initialization is used. Accordingly, we use the ImageNet dataset to pre-train the model due to the similarity of basic characteristics between ImageNet and our dataset. The pre-training process can greatly accelerate the convergence. For the RNN part, we pre-train the model without connection to the CNN layer. To realize the training, Tensorflow is used to implement the models and comparative experiments. The whole model is optimized by Adam optimizer. During training, the batch size is 64, the learning rate is 0.00001, and the epoch is 50. All the experiments are conducted on NVIDIA GeForce RTX 2080Ti.

Experimental setup

The method is evaluated on 3 image sets: FROSI (Foggy Road Sign Images) [53], FRIDA (Foggy Road Image Dataset) [54] and RMID (Real Multiview Images Dataset). FROSI and FRIDA are two synthetic datasets. FROSI contains images of simple road scenes and traffic signs, while FRIDA contains images of the urban road scenes under different weather conditions. The original image is blurred by adding the synthetic fog effects. And every image generates four synthetic images. 70% images of the two datasets are selected as the training set and the other images are used for testing.

Table 1 Three benchmark imagesets used in our experiment

Dataset	Blurry	Sub-blurry	Sub-clear	Clear
FROSI	41.8%	22.1%	10.9%	25.2%
FRIDA	11.4%	44.5%	33.7%	10.4%
RMID	10.8%	23.4%	34.0%	31.8%

RMID is a real multi-view image dataset collected by ourselves, which includes 3000 images. These images are grouped according to their capturing time and place. Every group of images only has different camera parameters. The visibility is labeled based on the report of the weather stations, ranging from 1km to 12km. The precision of the visibility is 0.1km. Thus, there are 111 visibility levels. Due to the rare situation of heavy fog, the distribution of visibility is imbalanced. Figure 6 shows several images in RMID. When using RMID to train our deep model, we randomly select 100 images for every visibility level to create a uniform distributed training set, and the other images form the test set.

We divide the three image sets by equally classifying the visibility into four grades: blurry, sub-blurry, sub-clear, and clear. As shown in Table 1, the distributions of the three image sets are very different.

To compare the performance of different methods, we use the mean absolute error (MAE) as the evaluation index, which is defined as:

Table 2 Results of different label distribution

Distribution	FROSI	RMID
Average Distribution	14.5%	19.3%
Triangle Distribution	15.0%	17.9%
Gauss Distribution ($\sigma^2=2.5$)	14.2%	17.2%
Gauss Distribution ($\sigma^2=2.0$)	13.6%	16.8%
Gauss Distribution ($\sigma^2=1.5$)	12.1%	15.8%
Gauss Distribution ($\sigma^2=1.0$)	13.3%	16.1%
Gauss Distribution ($\sigma^2=0.5$)	15.7%	20.4%

The best performance indicators are marked as bold

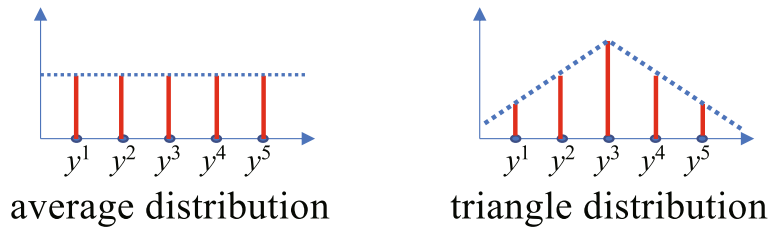


Fig. 7 The different label distributions: average distribution and triangle distribution

$$MAE = \frac{1}{n} \sum_{i=1}^n \frac{|y_i - \hat{y}_i|}{y_i} \tag{9}$$

where, y_i and \hat{y}_i are the ground truth and the predicted visibility of the i th testing image respectively, n is the number of the testing images.

Distribution analysis

We evaluate our choice of Gauss distribution and the parameter of the standard deviation σ . To study the influence of the distribution type, we compare the performance of Gauss distribution, average distribution, and triangle distribution. Figure 7 shows the shapes of the other two distributions. To search the optimal standard deviation σ , we run our method with 5 different values. Figure 8 shows the shapes of the gauss distribution with different σ . To simplify the comparison, we remove the prediction fusion and measure the accuracy of visibility estimation for a single image.

As shown in Table 2, different distributions have a different impact on the prediction results. From the table, we can see that the optimal distribution is the Gauss distribution. Moreover, the value of the standard deviation has a significant effect on the final performance. As shown in Fig. 8, if the value is too large, the Gauss distribution is very similar to the average distribution. If the value is too small, the Gauss distribution is over concentrated, which degenerates into the absolute label. Accordingly, we unify $\sigma^2 = 1.5$ in the following experiment since it leads to the best performance.

CNN-RNN analysis

We then explore the influence of the hyper-parameter K in the RNN module. This parameter controls the information exchange between CNN and RNN. To select the optimal value, we run our methods on the set FROSI with three different values. The experimental setting is without the prediction fusion stage, which is as the same as the distribution analysis experiment. The results are shown

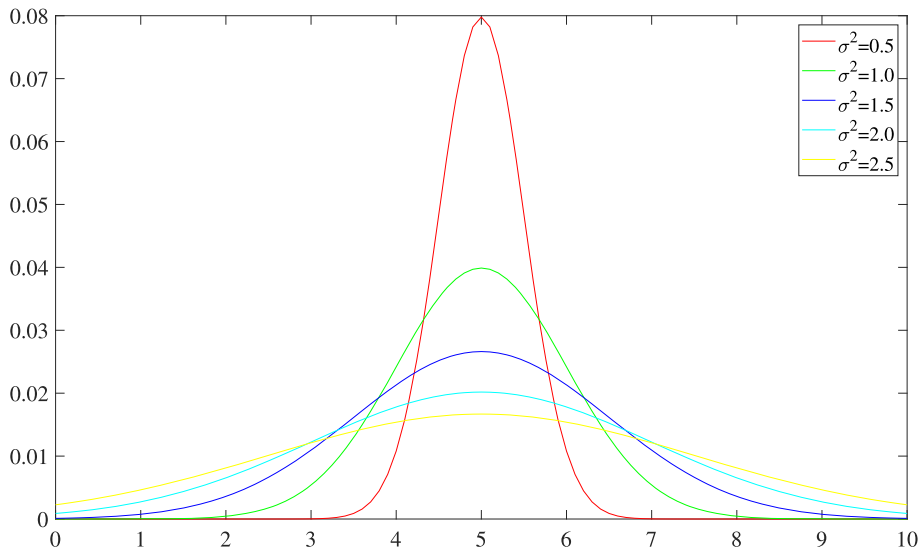


Fig. 8 Different σ for gauss distribution

Table 3 Results of different parameter K

$K=2$	$K=3$	$K=4$
16.7%	12.1%	15.1%

The best performance indicator is marked as bold

in Table 3. From the table, we can see that the effect of this parameter is relatively stable. In the following experiments, we set the parameter K as 3 since the performance is best.

Comparison with state-of-the-art methods

Figure 9 shows the test results of some images in the dataset, including the real value and prediction value. It is found that our result is very close to the ground truth. We also measure the timing of each step for processing the images in the cloud environment. During the training session, it takes 257ms average to update the parameters of the deep model for one image. To achieve model learning, it takes about 3 hours to train 2000 images. During the inference session, it costs 62ms average to estimate the visibility of every single image, and 55ms for prediction fusion. We also measure the network transfer time with the bandwidth 1000 Mbps. The size of every image is about 900 kb on average, thus the transfer delay is about 7ms. As our prediction fusion takes 4 images as input, the whole inference time with cloud computing is $62ms + 55ms + 4 \times 7ms = 145ms$ (the 4 images are estimated in parallel through the deep network). This satisfies the requirement of the real-time monitoring application. We also try to deploy the model in Huawei Atlas 200. After the test, it takes about 214ms average to estimate the visibility of one image, and 275ms for prediction fusion. Thus, the whole inference time is $214ms + 275ms = 489ms$ when using the programmable camera, which is 3.4 times that of cloud computing. This shows that cloud computing

is a superior choice in the application scenarios of visibility estimation.

Table 4 reports the comparisons of our method and previous state-of-the-art methods on three datasets from the view of prediction accuracy. According to the results in Table 4, our method achieves the best results. By analyzing the estimation results, we can see that label distribution is more effective than the absolute label or the ranking label when the network architecture is the same.

Ablation study

To prove the effectiveness of different designs, we compare our method with all components and alternatives with one of our choices disabled. We run the following variants:

No CNN - We remove the CNN module from the network.

No RNN - We remove the RNN module from the network.

No LDL - We use the absolute label instead of the distribution and turn it into a regression problem.

No Fusion - We use only one image for visibility estimation instead of fusing the predictions from different views.

We run our method and the variants on the three image sets. As shown in Table 4, we can see that our method achieves a significant performance boost compared with other variants. Among all the choices, the CNN module plays the most important role. The LDL, fusion, and RNN modules can also boost performance significantly. This demonstrates our contributions for visibility estimation.

We also compare different prediction fusion methods, such as average fusion, voting fusion, and max fusion. The average fusion method takes the average score of multi-view images as the output. The max fusion method chooses the visibility with the maximum score across the



Fig. 9 The estimation results of some images in RMID

Table 4 Comparisons with state-of-the-arts and ablation study

Methods	Label type	FROSI	FRIDA	RMID
VGG16 [55]	absolute	33.0%	26.4%	39.8%
ResNet [56]	absolute	26.3%	25.7%	30.2%
CNN+GRNN [7]	absolute	23.8%	29.1%	35.6%
DHCNN [57]	absolute	28.6%	29.2%	41.9%
VisNet [8]	absolute	21.4%	22.9%	27.5%
Relative CNN-RNN [16]	ranking	14.5%	13.8%	18.3%
Ours (without CNN)	distribution	23.3%	30.0%	33.6%
Ours (without RNN)	distribution	14.6%	16.7%	18.1%
Ours (without LDL)	absolute	14.0%	15.1%	16.3%
Ours (without Fusion)	distribution	12.1%	13.7%	15.8%
Ours (average fusion)	distribution	11.3%	11.4%	13.9%
Ours (voting fusion)	distribution	11.5%	12.0%	13.3%
Ours (max fusion)	distribution	10.8%	9.9%	14.7%
Ours	distribution	9.7%	8.9%	11.6%

The best performance indicator is marked as bold

multi-view images as the result. The voting fusion method uses the voting strategy to combine the predictions. Every image votes one visibility value with the highest probability. The visibility value with the maximum voting is the output of the fusion. As shown in Table 4, we can see that our prediction fusion method achieves the best performance.

Conclusion

We observe that image-based visibility estimation cannot successfully learn precise models when the labels are ambiguous. To solve this problem, we propose a deep label distribution learning method for visibility estimation. In our method, the visibility of every image is annotated by a label distribution. To learn from such annotation, we integrate CNN, RNN, and CPNN into a unified method, which locates the farthest region in the image and minimizes the difference between the predicted distribution

and the ground-truth distribution simultaneously. To realize actual engineering for real-time visibility monitoring, we combine cloud computing and our visibility estimation into a whole framework. The experiment shows that compared with the absolute label or ranking label, label distribution can achieve the best performance for visibility estimation. Meanwhile, our method can obtain visibility from the image efficiently.

Limitation and future work

The robustness and effectiveness of our method have been demonstrated by extensive experiments. It also has some limitations in some special cases. Figure 10 shows two cases. Our method relies on the information of the farthest region. If the region is not notable, the visibility may not be well predicted. Moreover, the backlighting and the local strong light still disturb the prediction. Our method can be improved in several directions in the future. Cur-



Fig. 10 Two failure cases of our method

rently, model learning and prediction fusion are separate. A future direction is to combine deep multi-view learning [58] into the training session, which leads to an end-to-end multi-view visibility prediction framework. Another direction is integrating edge computing [1, 14] to construct a more efficient and robust visibility monitoring framework.

Acknowledgments

We sincerely thank the reviewers and the Editor for their valuable suggestions.

Authors' contributions

MS and QL conceived and designed the study. MS and XH performed the simulations. MS, XH and XFL wrote the paper. All authors reviewed and edited the manuscript. All authors read and approved the final manuscript.

Funding

This work was supported by National Natural Science Foundation of China (61906036, 42075139), the Open Research Project of State Key Laboratory of Novel Software Technology (Nanjing University) (KFKT2019B02), the Fundamental Research Funds for the Central Universities (2242021k30056).

Availability of data and materials

The data used to support the finding of this study are available from the corresponding author upon request.

Declarations

Competing interests

The authors declare that they have no competing interests.

Author details

¹the School of Computer Science and Engineering, Southeast University, Nanjing, China. ²the Key Lab of Computer Network and Information Integration (Ministry of Education), Southeast University, Nanjing, China. ³Web Mining Laboratory, Department of Media and Communication, City University of Hong Kong, Hong Kong SAR, China. ⁴College of Meteorology and Oceanography, National University of Defense Technology, Changsha, China.

Received: 28 June 2021 Accepted: 17 August 2021

Published online: 28 August 2021

References

1. Xu X, Fang Z, Qi L, Zhang X, He Q, Zhou X (2021) Tripres: Traffic flow prediction driven resource reservation for multimedia iov with edge computing. *ACM Trans Multimed Comput Commun Appl* 17(2). <https://doi.org/10.1145/3401979>
2. van Rossum MCW, Nieuwenhuizen TM (1999) Multiple scattering of classical waves: microscopy, mesoscopy, and diffusion. *Rev Mod Phys* 71:313–371. <https://doi.org/10.1103/RevModPhys.71.313>
3. Mabrouki J, Azroul M, Fattah G, Dhiba D, Hajjaji SE (2021) Intelligent monitoring system for biogas detection based on the internet of things: Mohammedia, morocco city landfill case. *Big Data Min Analytics* 4(1):10–17. <https://doi.org/10.26599/BDMA.2020.9020017>
4. Babari R, Hautière N, Dumont É, Brémond R, Paparoditis N (2011) A model-driven approach to estimate atmospheric visibility with ordinary cameras. *Atmos Environ* 45(30):5316–5324. <https://doi.org/10.1016/j.atmosenv.2011.06.053>
5. Babari R, Hautière N, Dumont E, Papelard J-P, Paparoditis N (2011) Computer vision for the remote sensing of atmospheric visibility. In: *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, pp 219–226. <https://doi.org/10.1109/ICCVW.2011.6130246>
6. Li Q, Li Y, Xie B (2019) Single image-based scene visibility estimation. *IEEE Access* 7:24430–24439. <https://doi.org/10.1109/ACCESS.2019.2894658>
7. Li S, Fu H, Lo W (2017) Meteorological visibility evaluation on webcam weather image using deep learning features. *Int J Comput Theory Eng* 9:455–461
8. Palvanov A, Cho Y (2019) Visnet: Deep convolutional neural networks for forecasting atmospheric visibility. *Sensors* 19(6):1343
9. Qiu J, Chen C, Liu S, Zhang H-Y, Zeng B (2021) Slimconv: Reducing channel redundancy in convolutional neural networks by features recombining. *IEEE Trans Image Process* 30:6434–6445. <https://doi.org/10.1109/TIP.2021.3093795>
10. Liu J, Zhuang B, Zhuang Z, Guo Y, Huang J, Zhu J, Tan M (2021) Discrimination-aware network pruning for deep model compression. *IEEE Trans Pattern Anal Mach Intell*:1–1. <https://doi.org/10.1109/TPAMI.2021.3066410>
11. Mahmud MS, Huang JZ, Salloum S, Emara TZ, Sadatdiyov K (2020) A survey of data partitioning and sampling methods to support big data analysis. *Big Data Min Analytics* 3(2):85–101. <https://doi.org/10.26599/BDMA.2019.9020015>
12. Chen N, Wang Z, He R, Jiang J, Cheng F, Han C (2021) Efficient scheduling mapping algorithm for row parallel coarse-grained reconfigurable architecture. *Tsinghua Sci Technol* 26(5):724–735. <https://doi.org/10.26599/TST.2020.9010035>
13. Azroul M, Mabrouki J, Guezzaz A, Farhaoui Y (2021) New enhanced authentication protocol for internet of things. *Big Data Min Analytics* 4(1):1–9. <https://doi.org/10.26599/BDMA.2020.9020010>
14. Xu X, Wu Q, Qi L, Dou W, Tsai S-B, Bhuiyan MZA (2021) Trust-aware service offloading for video surveillance in edge computing enabled internet of vehicles. *IEEE Trans Intell Transp Syst* 22(3):1787–1796. <https://doi.org/10.1109/TITS.2020.2995622>
15. Koenderink J, J (1998) Pictorial relief. *Phil Trans R Soc A* 356(1740):6–6
16. You Y, Lu C, Wang W, Tang C-K (2019) Relative cnn-rnn: Learning relative atmospheric visibility from images. *IEEE Trans Image Process* 28(1):45–55. <https://doi.org/10.1109/TIP.2018.2857219>
17. Geng X (2016) Label distribution learning. *IEEE Trans Knowl Data Eng* 28(7):1734–1748. <https://doi.org/10.1109/TKDE.2016.2545658>
18. Busch C, Debes E (1998) Wavelet transform for analyzing fog visibility. *IEEE Intell Syst Appl* 13(6):66–71. <https://doi.org/10.1109/5254.736004>
19. Jourlin M, Pinoli J-C (1988) A model for logarithmic image processing. *J Microsc* 149(1):21–35. <https://doi.org/10.1111/j.1365-2818.1988.tb04559.x>
20. Bronte S, Bergasa LM, Alcantarilla PF (2009) Fog detection system based on computer vision techniques. In: *2009 12th International IEEE Conference on Intelligent Transportation Systems*, pp 1–6. <https://doi.org/10.1109/ITSC.2009.5309842>
21. Graves N, Newsam S (2011) Using visibility cameras to estimate atmospheric light extinction. In: *2011 IEEE Workshop on Applications of Computer Vision (WACV)*, pp 577–584. <https://doi.org/10.1109/WACV.2011.5711556>
22. Xiang W, Xiao J, Wang C, Liu Y (2013) A new model for daytime visibility index estimation fused average sobel gradient and dark channel ratio. In: *Proceedings of 2013 3rd International Conference on Computer Science and Network Technology*, pp 109–112. <https://doi.org/10.1109/ICCSNT.2013.6967074>
23. Giyenko A, Palvanov A, Cho Y (2018) Application of convolutional neural networks for visibility estimation of cctv images. In: *2018 International Conference on Information Networking (ICOIN)*, pp 875–879. <https://doi.org/10.1109/ICOIN.2018.8343247>
24. Wang H, Shen K, Yu P, Shi Q, Ko H (2020) Multimodal deep fusion network for visibility assessment with a small training dataset. *IEEE Access* 8:217057–217067. <https://doi.org/10.1109/ACCESS.2020.3031283>
25. Li J, Lo WL, Fu H, Chung HSH (2021) A transfer learning method for meteorological visibility estimation based on feature fusion method. *Appl Sci* 11(3):997. <https://doi.org/10.3390/app11030997>
26. Lo WL, Chung HSH, Fu H (2021) Experimental evaluation of pso based transfer learning method for meteorological visibility estimation. *Atmosphere* 12(7):828. <https://doi.org/10.3390/atmos12070828>
27. Xu X, Liu Q, Zhang X, Zhang J, Qi L, Dou W (2019) A blockchain-powered crowdsourcing method with privacy preservation in mobile environment. *IEEE Trans Comput Soc Syst* 6(6):1407–1419. <https://doi.org/10.1109/TCSS.2019.2909137>
28. Hua Y, Zhao Z, Li R, Chen X, Liu Z, Zhang H (2019) Deep learning with long short-term memory for time series prediction. *IEEE Commun Mag* 57(6):114–119. <https://doi.org/10.1109/MCOM.2019.1800155>
29. Jin Y, Guo W, Zhang Y (2020) A time-aware dynamic service quality prediction approach for services. *Tsinghua Sci Technol* 25(2):227–238. <https://doi.org/10.26599/TST.2019.9010007>
30. Zixuan W, Qingchi Y, Zhihong Y, Yang W, Zhiming F (2021) Visibility prediction of plateau airport based on lstm. In: *2021 IEEE 5th Advanced*

- Information Technology, Electronic and Automation Control Conference (IAEAC) Vol. 5. pp 1886–1891. <https://doi.org/10.1109/IAEAC50856.2021.9391060>
31. Bi R, Liu Q, Ren J, Tan G (2021) Utility aware offloading for mobile-edge computing. *Tsinghua Sci Technol* 26(2):239–250. <https://doi.org/10.26599/TST.2019.9010062>
 32. Xu X, Zhang X, Gao H, Xue Y, Qi L, Dou W (2020) Become: Blockchain-enabled computation offloading for iot in mobile edge computing. *IEEE Trans Ind Informal* 16(6):4187–4195. <https://doi.org/10.1109/TII.2019.2936869>
 33. Wang F, Zhu H, Srivastava G, Li S, Khosravi MR, Qi L (2021) Robust collaborative filtering recommendation with user-item-trust records. *IEEE Trans Comput Soc Syst*:1–11. <https://doi.org/10.1109/TCSS.2021.3064213>
 34. Qi L, Hu C, Zhang X, Khosravi MR, Sharma S, Pang S, Wang T (2021) Privacy-aware data fusion and prediction with spatial-temporal context for smart city industrial environment. *IEEE Trans Ind Informal* 17(6):4159–4167. <https://doi.org/10.1109/TII.2020.3012157>
 35. Wang K (2020) Migration strategy of cloud collaborative computing for delay-sensitive industrial iot applications in the context of intelligent manufacturing. *Comput Commu* 150:413–420. <https://doi.org/10.1016/j.comcom.2019.12.014>
 36. Nazari Jahantigh M, Masoud Rahmani A, Jafari Navimirour N, Rezaee A (2020) Integration of internet of things and cloud computing: a systematic survey. *IET Commun* 14(2):165–176. <https://doi.org/10.1049/iet-com.2019.0537>. <https://ietresearch.onlinelibrary.wiley.com/doi/pdf/10.1049/iet-com.2019.0537>
 37. Nakamura EF, Loureiro AAF, Frery AC (2007) Information fusion for wireless sensor networks: Methods, models, and classifications. *ACM Comput Surv* 39(3):9. <https://doi.org/10.1145/1267070.1267073>
 38. Sun S, Lin H, Ma J, Li X (2017) Multi-sensor distributed fusion estimation with applications in networked systems: A review paper. *Inf Fusion* 38:122–134. <https://doi.org/10.1016/j.inffus.2017.03.006>
 39. Xiao F (2019) Multi-sensor data fusion based on the belief divergence measure of evidences and the belief entropy. *Inf Fusion* 46:23–32. <https://doi.org/10.1016/j.inffus.2018.04.003>
 40. Yager RR (2019) Generalized dempster–shafer structures. *IEEE Trans Fuzzy Syst* 27(3):428–435. <https://doi.org/10.1109/TFUZZ.2018.2859899>
 41. Majumder S, Pratihari DK (2018) Multi-sensors data fusion through fuzzy clustering and predictive tools. *Expert Syst Appl* 107:165–172. <https://doi.org/10.1016/j.eswa.2018.04.026>
 42. Wei W, Liang J (2019) Information fusion in rough set theory: An overview. *Inf Fusion* 48:107–118. <https://doi.org/10.1016/j.inffus.2018.08.007>
 43. Zhang L, Ding L, Wu X, Skibniewski MJ (2017) An improved dempster–shafer approach to construction safety risk perception. *Knowl Based Syst* 132:30–46. <https://doi.org/10.1016/j.knosys.2017.06.014>
 44. Razi S, Karami Mollaei MR, Ghasemi J (2019) A novel method for classification of bci multi-class motor imagery task based on dempster–shafer theory. *Inf Sci* 484:14–26. <https://doi.org/10.1016/j.ins.2019.01.053>
 45. Liu F, Zhao Q, Yang Y (2018) An approach to assess the value of industrial heritage based on dempster–shafer theory. *J Cult Herit* 32:210–220. <https://doi.org/10.1016/j.culher.2018.01.011>
 46. Chen Y, Kang Y, Chen Y, Wang Z (2020) Probabilistic forecasting with temporal convolutional neural network. *Neurocomputing* 399:491–501. <https://doi.org/10.1016/j.neucom.2020.03.011>
 47. Greff K, Srivastava RK, Koutnik J, Steunebrink BR, Schmidhuber J (2017) Lstm: A search space odyssey. *IEEE Trans Neural Netw Learn Syst* 28(10):2222–2232. <https://doi.org/10.1109/TNNLS.2016.2582924>
 48. Dey R, Salem FM (2017) Gate-variants of gated recurrent unit (gru) neural networks. In: 2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS). pp 1597–1600. <https://doi.org/10.1109/MWSCAS.2017.8053243>
 49. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In: *Proc. Adv. Neural Inf. Process. Syst.*, vol. 25. MIT press, Cambridge. pp 1097–1105
 50. Geng X, Yin C, Zhou Z-H (2013) Facial age estimation by learning from label distributions. *IEEE Trans Pattern Anal Mach Intell* 35(10):2401–2412. <https://doi.org/10.1109/TPAMI.2013.51>
 51. Johnson R, Zhang T (2013) Accelerating stochastic gradient descent using predictive variance reduction. In: *Proc. Adv. Neural Inf. Process. Syst.*, vol. 26. MIT press, Cambridge. pp 315–323
 52. Sarabi-Jamab A, Araabi BN (2018) How to decide when the sources of evidence are unreliable: A multi-criteria discounting approach in the dempster–shafer theory. *Inf Sci* 448–449:233–248. <https://doi.org/10.1016/j.ins.2018.03.001>
 53. Tarel J-P, Hautière N, Cord A, Gruyer D, Halmaoui H (2010) Improved visibility of road scene images under heterogeneous fog. In: *Proc. IEEE Intell. Veh. Symp.* pp 478–485. <https://doi.org/10.1109/IVS.2010.5548128>
 54. Belaroussi R, Gruyer D (2014) Impact of reduced visibility from fog on traffic sign detection. In: *Proc. IEEE Intell. Veh. Symp.* pp 1302–1306. <https://doi.org/10.1109/IVS.2014.6856535>
 55. Simonyan K, Zisserman A (2015) Very deep convolutional networks for large-scale image recognition. In: Yoshua B, Yann L (eds). *Proc. Int. Conf. Learn. Represent.*, San Diego
 56. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.* pp 770–778. <https://doi.org/10.1109/CVPR.2016.90>
 57. Palvanov A, Im Cho Y (2018) Dhcn for visibility estimation in foggy weather conditions. In: *Proc. Joint 10th Int. Conf. Soft Comput. Intell. Syst. (SCIS) 19th Int. Symp. Adv. Intell. Syst. (ISIS)*. pp 240–243. <https://doi.org/10.1109/SCIS-ISIS.2018.00050>
 58. Yan X, Hu S, Mao Y, Ye Y, Yu H (2021) Deep multi-view learning methods: A review. *Neurocomputing* 448:106–129. <https://doi.org/10.1016/j.neucom.2021.03.090>

Publisher's Note

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

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► [springeropen.com](https://www.springeropen.com)
