IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Detecting the Preceding Vehicles Under Foggy Weather

[1]Ch.Lakshmana, [2]D.V.Sushma , [3]Sk.Davood Ibrahim , [4]Sk.Shabul , [5]B.Jaswanth

DVR & Dr. HS MIC College of Technology

***Abstract*—** This research suggests a visibility detection approach based on the identification of the taillight signals of the previous vehicle using vehicle pictures. We first construct the system for picture enhancement before identifying the vehicle from the enhanced image. We are building a system that primarily serves as a means of vehicle identification without using the taillights. The other is to recognize moving vehicles using taillight segmentation. Utilising a YOLOv2 detector network, automobiles are detected in this work. The enormous variety in picture appearance and imaging interference mean that even though numerous automated detection approaches, including those based on image processing techniques, have been developed, the detection performance can still be improved.

Keywords: Atmospheric haze removal, Image Processing, YOLOV2, MATLAB.

I. INTRODUCTION

The maximum horizontal distance at which a target is unmistakably visible to a person with normal eyesight is known as visibility.There has been a lot of critical attention paid to the subject of visibility detection in foggy situations. The main visibility detection techniques now in use on foggy days are instrument measurement and visual measurement. Among these, the first has stringent standards for surveyors, is challenging to spread, and cannot be relied upon for accuracy. When using transmission or scattering measuring equipment, the instrument measurement approach is commonly employed to check air visibility, but both have drawbacks in specific application scenarios that make them undesirable. Furthermore, it is challenging to apply the concept to highway networks because to issues with a limited service life and high maintenance expenses. Foggy days may be accompanied by mass fog, a type of fog with decreased visibility that can form in an area with a range of tens to hundreds of metres during dense fog. It has a clear line of sight from the outside, but the inside is cloudy. Mass fog is extremely localised and challenging to forecast, particularly on highways.

In many applications, vehicle detection is a crucial task. Vehicle detection aims to recognise the presence of automobiles in a given image and precisely pinpoint their locations. This job entails identifying the various components of a car, such as the wheels, chassis, and windows, and separating them from other things in the environment.

Vehicle detection is one of the essential elements of the overall system in this article. Using a combination of computer vision and deep learning methods, the system seeks to detect. The technology will be able to recognise various vehicle types, including cars, lorries, and buses, and follow their movements over time.

In order to identify the vehicles in the image, we will use an object identification technique like YOLO. These systems accurately detect the presence of automobiles by utilising deep learning architectures and neural networks. In order to preprocess the image frames and increase the precision of the detection algorithms, we will also employ image processing techniques such thresholding, edge detection, contour detection, and image

enhancement. The technique of determining the degree of visibility in a certain scene or image is known as visibility detection. The degree of clarity and detail that can be seen in an image is referred to as visibility. Fog, rain, smoke, haze, dust, and other elements can all have an impact on visibility, which can substantially lower the quality of an image and make it challenging to see crucial details.

Visibility detection is a critical problem in computer vision that is used in many applications, including as surveillance, autonomous driving, and robots. Visibility detection seeks to estimate the visibility level within a specific picture or video frame. Other computer vision algorithms, such object detection, tracking, and recognition, which depend on clear, detailed images, can benefit from this knowledge by performing better.

The visibility detection process involves analyzing various image features such as color, contrast, and texture to estimate the level of visibility. Various techniques such as image enhancement, dehazing, and deblurring can be used to improve the visibility level in the image or video frames.

In this paper, visibility detection is an essent ial component of the system, and it will be used to enhance the performance of other computer vision algorithms. The visibility detection module will utilize state-of-the-art techniques such as image enhancement and dehazing to estimate the level of visibility in the image or video frames. The output of the visibility detection module will be used by other modules in the system to improve their accuracy and reliability.

In computer vision, visibility detection is a crucial task that is essential to many applications. In this paper, visibility detection will be used to enhance the performance of other computer vision algorithms and provide accurate and reliable results.

II. EXISTING METHOD

Mask R-CNN, which was constructed on top of Faster R-CNN, is the segmentation model that is the most sophisticated. The Faster R-CNN produces bounding boxes for each object and a class label with a confidence score because it is a region-based convolutional neural network.

Let's first examine the Faster R-CNN architecture, which consists of two phases, in order to comprehend Mask R-CNN.

Stage1: The region proposal network and the backbone network (ResNet, VGG, Inception, etc.) make up the first phase. These networks execute once for each image to produce a list of recommended areas. The feature map's suggested regions are those where the object can be found.

Stage2: In stage 2, Each recommended region that was obtained had bounding boxes and item categories predicted by the network. A constant-size vector must always be used by fully connected layers in networks to provide predictions, even though the size of each recommended region may differ.The RoIAlign method, also known as the RoI pool and very similar to MaxPooling, is used to determine the sizes of these suggested regions.

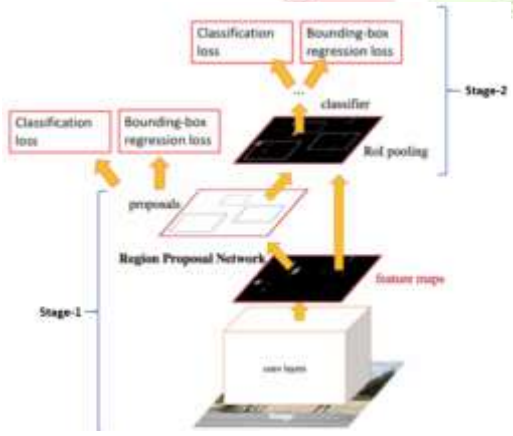


Fig 1: A single, unified network for object detection is faster R-CNN.

Using R-CNN, bounding box and object class predictions may be made more quickly. The quicker R-CNN now has a branch for segmentation mask prediction for each Region of Interest (RoI). In the second stage of Faster R-CNN, the RoIAlign approach is employed to maintain spatial information that falls out of alignment when using RoI pool. Binary interpolation is utilised, RoIAlign creates a fixed-size feature map, such as a 77map.

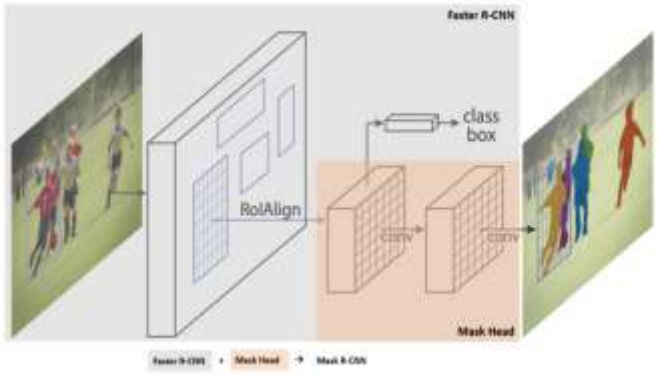


Fig 2: The Mask R-CNN framework for instance segmentation

The Mask head, which contains two convolution layers, receives the output from the RoIAlign layer next. It segments an image pixel by pixel, creating a mask for each RoI. CNN can be trained to tell us which aspect of the image had the greatest influence on it. To do this, ask CNN to indicate a box around the object. This task, called object detection in deep learning parlance, is rather easy to complete. We must first use a tool to construct bounding boxes around photographs when preparing our data. This is a relatively easy approach when using free web resources. The final/output layer of the CNN is then transformed into a softmax layer with 4 + k outputs, which comprises the bounding box's x, y, height, and width as well as class probability scores for k classes.

You could be wondering why we selected challenging ideas to comprehend, like height and breadth and x and y coordinates. Why can't we just determine each box corner's (x,y) coordinates? Yes, we can, however to represent the box, we'll need to learn a total of 8 variables assuming we only discover 4 sets of variable pairings.But if we use this strategy, we just need 4 employees.

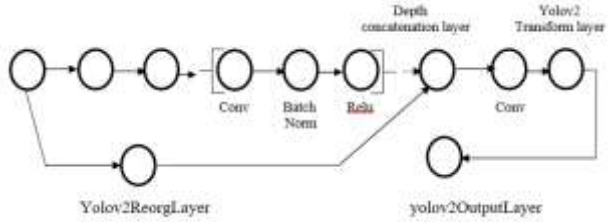
The fascinating topic of semantic segmentation is up for discussion. Once more, this is just a fancy way of saying that you are simply colouring in an image, much like you would in a colouring book for kids. Typically, instance segmentation can be solved in two steps:

1. To construct bounding boxes around each member of a class, use an object detection approach.
2. Each bounding box should undergo semantic segmentation.

This incredible, straightforward concept actually functions rather effectively. It works because step 2's semantic segmentation is given a set of photos that is certain to contain only one instance of the primary class, assuming that step 1 is highly accurate. Predicting which pixels in a picture belong to the primary class (a cat, dog, or person, for example) and which pixels belong to the background is the model's job in step two.

Another exciting aspect is that instance segmentation will be essentially complete if we can address the separate issues of numerous bounding boxes and semantic segmentation. The good news is that practical solutions have been developed for each of these problems, and putting them together is a rather straightforward procedure.

The Mask R-CNN operating system is quite straightforward once more. In terms of deep learning research, all that was



accomplished was the combination of two cutting-edge models and some linear algebra experiments. A region proposal network (RPN) and a binary mask classifier make up the majority of the model. Getting a set of bounding boxes that might include a significant object is the first step. RoI Align is the slang term now in use. The RoI Align network produces a number of potential bounding boxes rather than a single distinct one, in accordance with the object detection principles (explained above, in case you forgot!). Another model—which we won't discuss here—is used to improve these boxes. Information on the RoI Align network in more detail.

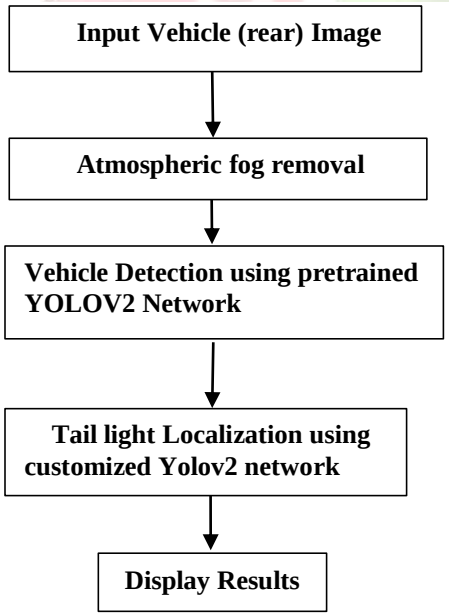
Disadvantages:

• The feature map for the region proposal takes up a lot of disc space;
• RCNN and Fast RCNN take time to accurately identify objects in a picture on the GPU.
• Training loss in Faster RCNN will be substantial.

III. PROPOSED METHOD

Here, we demonstrate a technique for enhancing photos by removing haze, detecting vehicles, and detecting tail lights. Some vehicles still don't turn on their taillights when it's foggy outside, despite the majority doing so. For the visibility rating approach used in this work to be successful for all cars, it is important to be able to distinguish between individual vehicle taillights.

Here, we are going to discuss the Yolo network.



Block Diagram of Proposed Method

Layers of YOLOv2 Network

By hovering over a block in YOLOv2, you can learn more about it. With the final convolution block being the exception, each convolution block is normalised using BatchNorm before Leaky Relu is activated. YOLO creates an SS grid from the provided image. Each grid cell predicts a single object. One example is the "person" item with a blue dot at its centre in the yellow grid cell below. There will be a predetermined number of border boxes in each grid cell. The yellow grid cell in this picture forecasts two boundary boxes (the blue boxes), which show the person's position. The one-object rule, however, limits the proximity of recognised items.

Each grid cell in,
Regardless of the number of boxes B, it recognises a single object, provides a box confidence score to each border box and forecasts conditional class probabilities C (one for every object class and its likelihood class).

The box confidence score is kept for the boundary boxes. The confidence score is based on the bounding box's precision and likelihood of containing an object (or being empty). You may get the image's height and width by subtracting the bounding box. the resulting numbers, and they are all between 0 and 1. Twenty conditional class probabilities are present in each cell. The conditional class probability (one probability per category for each cell) represents the potential that the recognised object belongs to a particular class.

Box confidence score multiplied by conditional class probability is the class confidence score.
The localization (where an object is placed) and categorization confidence are both measured. Those terminology for score and probability are readily conflated. For your future reference, the mathematical definitions are provided below.
Box confidence score is the product *of P*r (object) and The probability of an IoU conditional class is equal to Pr (class|object).
Pr (classi) IoU, or a box confidence score conditional class probability, is the definition of class confidence.

The probability that an object is in the box is expressed as Pr(object).
The intersection over union (IoU) of the actual data and the predicted box is referred to as IoU.

The likelihood that an item belongs to a class given its existence is known as Pr(class|object).
The likelihood that an object belongs to a class is expressed as Pr(classi).i
Each grid cell's boundary boxes are predicted using YOLO. To determine the loss for real positive, we just want to hold one of them accountable. As a result, we opt for the alternative that most closely intersects with reality (IoU). This strategy has increased the level of specialisation in the bounding box forecasts. With each projection, exact sizes and aspect ratios become more precise.

The loss functh ion is made up of:
• the erasure of classification.
• the differences between the actual boundary box and the localization loss, often known as the anticipated border box.
• the drop in self-assurance (as a result of the box's impartiality).

Classification loss: If an item is discovered, The class conditional probability squared error for each class in each cell represents the classification loss in that cell.

$$\Sigma_{i=0}^{s^2} 1_i^{obj} \Sigma_{c \varepsilon classes} (X_i(c) - \widehat{X}_i(c))^2$$

1_i^{obj} = 1 if a cell i contains an object; otherwise, 0.

The localization loss is used to calculate the sizes and positions of the projected border boxes. We only count the boxes that can detect the object.

$$\lambda_{coord} \Sigma_{i=0}^{s^2} \Sigma_{j=0}^{B} 1_i^{obj} [(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2]$$
$$+\lambda_{coord} \Sigma_{i=0}^{s^2} \Sigma_{j=0}^{B} 1_i^{obj} [(\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2]$$

1_{ij}^{obj} = 1 If cell i's jth boundary box is in charge of finding the object, then _coord should be increased to account for the loss in boundary box coordinates.

Absolute mistakes shouldn't be treated equally in large and small boxes. In other words, a 1-pixel error in a small box is equivalent to a 2-pixel mistake in a large one. By predicting the bounding box's height and breadth as their square roots rather of their actual values, Yolo partially resolves the problem. We additionally increase the loss by coordinate to emphasise the border box precision even more.

Image Input Layer: Specify the layer name to build a layer for image input. Images are fed into the system using the normalisation layer of an image input network. To specify the image size, use the input Size option. The size of an image is influenced by the quantity of colour channels as well as their width and height.

Convolutional layer: The input is processed using a sliding convolutional filters are used in a 2-D convolutional layer. Using the 2-D convolutional layer Convolution2dLayer may be produced. One or more components make up the convolutional layer.A convolutional layer's Neurons make up filters, stride, and other things connected to specific areas of the input images or the outputs of the layer before. The layer looks for things in the photos that are localised by these zones. You can control the dimensions of these zones when making a layer with the convolution2dLayer function by using the filter Size input parameter.

Dilated Convolution: A dilated convolution happens when there are greater gaps between the filter's pieces. To give the dilation factor, use the 'Dilation Factor' element. Without altering the parameters or processing capacity, The area of the input that the layer can view is increased by using dilated convolutions. The layer enlarges the filters by adding zeros between each filter element.Greater gaps between the filter's parts result in a dilated convolution. To give the dilation factor, use the 'Dilation Factor' element. Without altering the parameters or processing power, The layer's receptive field, or the area of the input that it can view, is widened using dilated convolutions. By placing zeros between each filter element, the layer makes the filters larger.

Feature Maps: To create a feature map, the convolution is given the same bias and weights as the filter. With specific bias and weights, a convolution is used to produce each feature map. There are equal numbers of feature maps and filters. A convolutional layer's total number of parameters is calculated using the formula ((h*w*c + 1)*Number of Filters), where 1 denotes the bias.

Zero Padding: You can also apply zero padding to input picture borders on both the horizontal and vertical axes by using the 'Padding' name-value pair option.The procedure of padding involves increasing the margins of a photo input by adding zeros in rows or columns. The padding can be changed in order to change the layer's output size. Using a 1-pixel padding, In this image, a 3-by-3 filter examines the input. The top map displays the result, while the lower map displays the input.

Rectified Linear Unit (ReLU): Rectified Linear Units make up the non-linear activation function. A ReLU will urge you to round the value up to zero if it is less than zero. creating a ReLU layer with reluLayer. A ReLU layer's threshold operation, which changes any value less than zero to zero, is applied to each input element. A batch normalisation layer, a convolutional layer, and a rectified linear unit (ReLU), which is specified by a ReLU layer, are typically placed after a ReLU layer.While maintaining the input's original size, a ReLU layer performs threshold operations on each element and sets all values lower than zero to zero. Additional nonlinear activation layers occasionally enhance network accuracy by performing a range of tasks. A list of activation layers may be found under Activation Layers.

Batch normalization layer: Batch normalisation significantly speeds up training, improves network predictability, and lowers over fitting through regularisation. Batch normalisation comprises normalising the scope activation layer of the current batch and dividing by the standard deviation of the activations after subtracting the batch mean. A batch normalisation layer might be produced using this technique. Each input channel is normalised in a mini-batch using a batch normalisation layer. To speed up the training of convolutional neural networks and reduce their sensitivity to initialization, use batch normalisation layers between convolutional layers and nonlinearities, such as ReLU layers.

Prior to update the parameters as learnable parameters during network training.A learnable scaling factor and learnable offset are then applied to the input. Reduce the regularisation of dropouts and L2 if you'd like. When utilising batch normalisation layers, the training activations for a certain image according to which other images just so happen to be in the same mini-batch. To increase the regularising effect, Before each training cycle, consider reorganising the training data.To control how often the data is shuffled during training, utilise the training options' 'Shuffle' name-value pair option.

Max and Average Pooling Layers: By dividing A max pooling layer achieves down-sampling by dividing the input into rectangular pooling regions and figuring out the maximum of each zone. To generate a typical pooling layer, use averagePooling2dLayer.The pooling layers are thus put before the convolutional layers, downsampling reduces the amount of connections to the subsequent layers. They don't pick up any knowledge, but they reduce the amount that levels below must study. Additionally, they help to reduce overfitting.

The best outcomes when utilising a max pooling layer are for the rectangular input regions. The maxPoolingLayer's pool Size input determines the dimensions of the rectangle's component parts. For instance, in a pool with a dimension of [2, 3], work best for the layer. A median pooling layer produces the median values of rectangular input regions. The dimensions of the rectangular components are determined by the pool parameter of the averagePoolingLayer. If the layer has sections with dimensions of 2 and 3, it will show their average value, for instance, the pool size is [2, 3].

Using the 'Stride' name-value pair argument, pooling layers do both horizontal and vertical analysis of the input in steps of the size you specify. If the size of the pool is less than or equal to the stride, the pooling zones do not overlap. In the case of n-by-n input and h-by-h pooling region size (Pool Size and Stride are equal), a single convolutional layer channel is subjected to a max or average pooling layer application results in an output that is n/h by n/h.

Dropout Layer: Utilise a to create one. An input element is randomly set to zero .T He rand dropout mask (size X) During training, probability is employed to randomly set input elements to zero, with X serving as the layer input. The final step is to scale the remaining input items by 1/ (1-Probability) by the layer. This approach successfully avoids over-fitting and modifies the basic topology of the network between iterations. During training, more components are deleted as the number grows. The output and input of the layer are the same at the moment of prediction. This layer has no learning, much like the maximum .

Fully Connected Layer: Build a fully connected layer using a fully linked layer. A fully connected layer multiplies the input by a weight matrix before adding a bias vector. One or more completely linked layers come after the convolutional (and downsampling) layers.Fully linked neurons in one layer communicate with all other neurons in the layer above them, as the name suggests. The properties required to classify the images in classification tasks are present in the final, fully linked layer. This explains why the data set's number of classes is identical to the network's final fully linked layer's output Size parameter. Issues with regression call for an equal distribution of output size and response variables.

By including the required name-value pair parameters, you can modify the regularisation and learning rate settings while generating the fully linked layer, it is possible to have a better understanding of the possibilities for global and layer training. Prior to being added to a completely linked layer with a bias vector, b, the input is multiplied by a weight matrix, W. The fully linked layer of an LSTM network responds differently at each time step if it receives a sequence as input. The fully linked layer generates an output array of size Z.Sizing by N and S Consider that the layer above it develops an array X with the following dimensions: D, N, and S. The Z component, which represents the time step t of time and the time step t of X, is absent.

Output Layers:

Softmax and Classification Layers: A softmax function is applied to the input using a softmax layer.To construct softmax layers, utilise softmax layers. A layer of categorization can be created using the classification layer. The final totally linked layer must be followed by a softmax layer and a classification layer in order to address classification-related problems.The normalised exponential, sometimes referred to as the softmax function**,** is the logistic sigmoid function's multi-class variation. In traditional classification networks, After the softmax layer, There has to be a classification layer added.

Regression Layer: Create a layer for regression by utilising the layer for regression. For issues involving regression, The loss of half-mean-squared-error is computed by a regression layer. a layer of regression is necessary for common regression issues come after the last fully linked layer.If R represents the overall number of responses, ti is the desired output, and yi is the predicted value for response i by the network, , serves as the loss The boundary boxes' box confidence score is preserved.

IV. ADVANTAGES AND APPLICATIONS

Advantages:

• Fast. A single network is used to make predictions (object locations and classes), which is advantageous for real-time processing. Accuracy can be totally taught to improve.

• YOLO is used more frequently. Moving from natural images to other domains, like artwork, it performs better than previous approaches.

• Classifiers can only be used in that location according to certain methods. When predicting limitations, YOLO uses the complete image. The additional information reduces the number of false positives in background regions for YOLO.

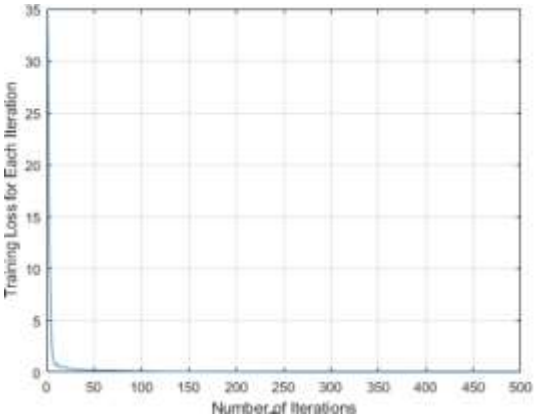
• In YOLO, a single item is detected in each grid cell. It is necessary to enforce spatial variety in order to create predictions.

Applications:

- Various fields:
- Object detection
- Person detection
- Vehicle Detection

V. RESULTS

Input Image



Training Loss

Fog Removed

Detected Image

Detected Vehicle

VI. FUTURE SCOPE OF STUDY

There are many interesting applications of this research in different fields

1.Autonomous vehicles: YOLOv2 could be used for object detection in autonomous vehicles, such as cars or drones, to detect and classify objects in real-time.

2.Surveillance: YOLOv2 could be used in surveillance systems to detect and track objects in real-time, which could help improve security and safety in public places.

3.Industrial automation: YOLOv2could be used in industrial automation to detect and classify objects in real-time, which could help improve efficiency and reduce errors in manufacturing and production processes.

4.Medical imaging: YOLOv2 could be used in medical imaging to detect and classify objects, such as tumors or abnormalities, which could help improve diagnosis and treatment of medical conditions.

Overall, YOLOv2 still has potential applications in various fields, and its future scopewill depend on the specific needs and requirements of different industries and use cases.

VII. CONCLUSION

In this paper, utilising deep learning and image processing, we proposed a method for detecting automobiles and tail lights. Two steps made up the algorithm's representation. It was trained to recognise automobiles and tail lights using a YOLO v2 model. The trained YOLO v2 achieved a very low in training loss and good accuracy.

VIII. REFERENCES

[1]. R. G. Hallowell, M. P. Matthews, and P. A. Pisano, ‘‘Automated extraction of weather variables from camera imagery,’’ in Proc. MCTRS, Ames, Iowa, Aug. 2005, pp. 1–13.

[2]. K. Mori, T. Kato, T. Takahashi, I. Ide, H. Murase, T. Miyahara, and Y. Tamatsu, ‘‘Visibility estimation in foggy conditions by in-vehicle camera and radar,’’ in Proc. 1st Int. Conf. Innov. Comput., Inf. Control (ICICIC), vol. 1. Beijing, China, Aug./Sep. 2006, pp. 548–551.

[3]. M. Gabb, S. Krebs, O. Lohlein, and M. Fritzsche, ‘‘Probabilistic inference of visibility conditions by means of sensor fusion,’’ in Proc. IEEE Intell. Vehicles Symp., Dearborn, MI, USA, Jun. 2014, pp. 1211–1216.

[4]. N. Hautiere, R. Babari, and E. Dumont, ‘‘Estimating meteorological visibility using cameras: A probabilistic model-driven approach,’’ in Proc. ACCV, Berlin, Germany, 2010, pp. 243–254

[5]. C. Zhang, M. Wu, J. Chen, K. Chen, C. Zhang, C. Xie, B. Huang, and Z. He, ‘‘Weather visibility prediction based on multimodal fusion,’’ IEEE Access, vol. 7, pp. 74776–74786, 2019.

[6]. K. W. Kim, ‘‘The comparison of visibility measurement between image based visual range, human eye-based visual range, and meteorological optical range,’’ Atmos. Environ., vol. 190, pp. 74–86, Oct. 2018.