Object Detection in Foggy Conditions by Fusion of Saliency Map and YOLO

[1]Sarthak Katyal
Department of Computer Engineering
Delhi Technological University
Delhi, India
sarthak.katyal1998@gmail.com

[2]Sanjay Kumar
Department of Computer Engineering
Delhi Technological University
Delhi, India
sanjay.kumar@dtu.ac.in

[3]Ronak Sakhuja
Department of Computer Engineering
Delhi Technological University
Delhi, India
ronaksakhuja@gmail.com

[4]Samarth Gupta
Department of Computer Engineering
Delhi Technological University
Delhi, India
samarthgupta1011@gmail.com

***Abstract* — Under foggy conditions, visibility decreases and causes many problems. Less visibility due to foggy conditions while driving increases the risk of road accidents. It is important to detect and recognize the nearby objects under such conditions and predict the distance of collision. There is a need to devise an object detection mechanism during foggy conditions. The paper proposes a solution to this problem by proposing a VESY(Visibility Enhancement Saliency YOLO) sensor which uses an algorithm that fuses the saliency map of the foggy image frame with the output generated from object detection algorithm YOLO (You Only Look Once). The image is sensed using image sensors in a stereo camera which are activated using a fog sensor and a depth map is generated to calculate the distance of collision. Dehaze algorithm is applied to improve the quality of the image frame whose Saliency image is generated on the basis of region covariance matrix. YOLO algorithm is also implemented on the improved quality image. The proposed fusion algorithm gives the bounding boxes of the union of the objects detected by Saliency Map and YOLO Algorithm thus proving to be a viable solution for real-time applications.**

Keywords — Saliency map, YOLO, depth map, dehaze, fusion.

I. INTRODUCTION

YOLO Architecture is a fully convolutional neural network which translates image pixels to coordinates of the generated bounding boxes and probabilities of class[7]. It is trained on full images for optimising detection. The network has a speed of 150 frames per second on Titan X GPU. Thus, YOLO can be used for streaming video in real time. YOLO has some limitations in predicting bounding boxes as every matrix element can only predict 2 boxes and can have 1 class thus limiting the number of objects nearby that can be predicted. Also, YOLO algorithm doesn't give the desired solution when working under severe conditions that affect image quality like foggy conditions. Foggy conditions make the image distorted and hence they are not detected by the algorithm. As shown in Fig. 1 (a) and (b), YOLO algorithm is unable to detect all the objects under foggy conditions. In Fig. 1 (a), One Vehicle was not detected by the general YOLO algorithm. Similarly, in Fig. 1 (b) many vehicles were left unidentified. This points to the fact that there is a need to enhance the existing image detection algorithm YOLO such that it can be used in foggy conditions.

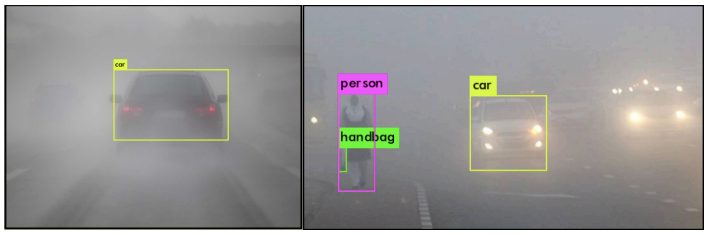


Figure 1 (a) and (b) : Failure of YOLO Detection Algorithm to distinct all objects

For a distorted frame, like an image in Fig. 1 (a), it is important to identify each pixel's unique quality. Thus, Saliency map is generated to represent the image in a more meaningful manner. It extracts the contours from the image and this tool of image segmentation highlights the objects that are not clear in the image frame in a better manner. It can be clearly shown that the objects that were not detected and recognized by YOLO in Fig. 1 (a) and (b) have clear and distinct boundaries in their respective Saliency maps. The Saliency Map of the images have been shown in Fig. 2 (a) and (b). The intensities of white shaded regions for each pixel the saliency map points to the fact that an object exists. These intensities are incorporated in the Saliency matrix.

Figure 2 (a) and (b) : Saliency Map of the Images used in Fig 1 (a) and (b)

978-1-5386-5147-6/18/$31.00 ©2018 IEEE

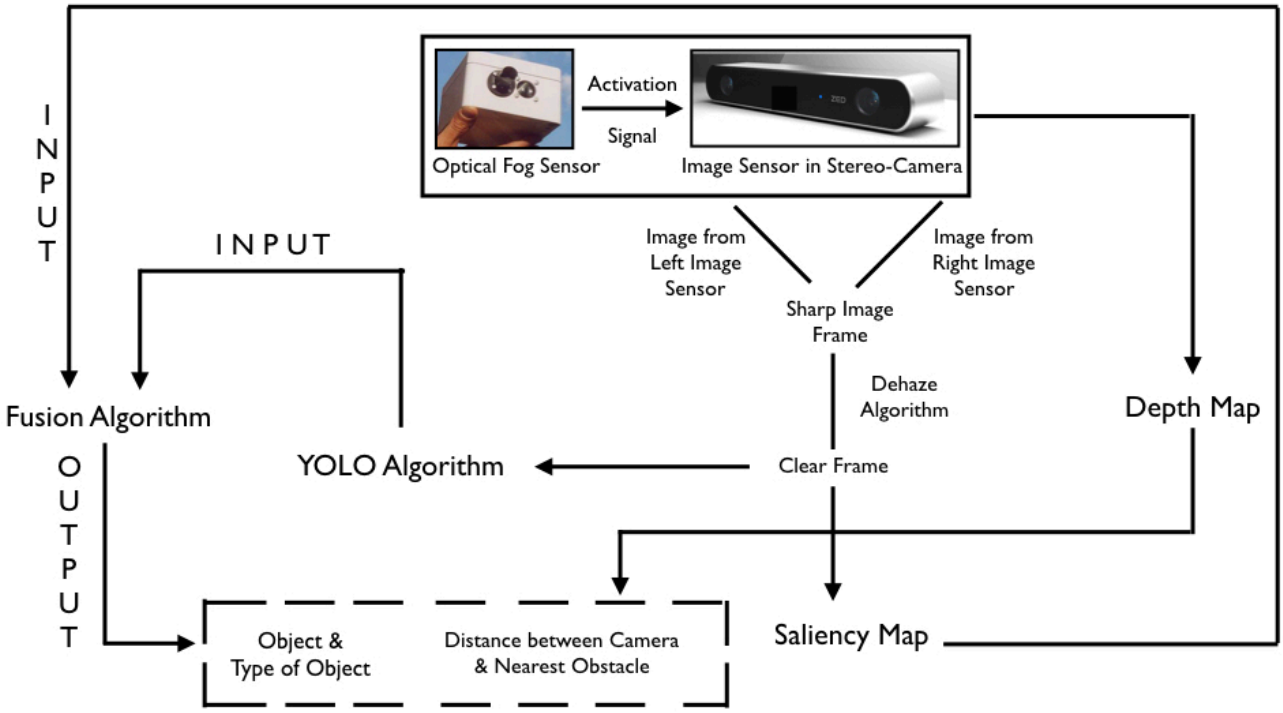


Figure 3: Prototype of VESY Sensor (Visibility Enhancement Saliency YOLO Sensor)

The results in Fig. 2 (a) and (b) thus provides additional proof to our notion of using Saliency Map for detecting objects that were not detected by YOLO algorithm. In this paper, we propose the VESY sensor which uses an algorithm to enhance the foggy image, by using the results yielded by the implementation of YOLO and the creation of saliency map for object detection and recognition during foggy conditions.

II. RELATED WORK

Reference [1] proposed a fast image defogging algorithm. We have incorporated this algorithm to obtain clear image frames. Reference [2] devised a recent method for salient object detection on images with less visibility. We have used the covariance feature matrix according to the attributes of images with fog in our proposed algorithm. The accuracy and adaptability of the method under foggy conditions is an advantage to our advanced study of fusion for better implementation of YOLO under such conditions. Reference [3] proposed a method for creation of a depth map from a stereo sensor for calculating the distance of collision. Reference [7] provides a detailed insight of You Only Look Once: Unified, Real-Time Object Detection Mechanism. The implementation fails to give desired results under adverse conditions and therefore our fusion algorithm optimises the result thus making YOLO more adaptive in nature and broader in usage to give the desired results.

III. PROPOSED WORK

A. Overview

This research yields a solution for object detection under foggy conditions on the sensed image and proposes the method to compute the distance between the object and the viewer. Fog sensor is used to activate the system and once the presence of an object is detected by our proposed algorithm, a bounding box is made around the object, and the type of the object is displayed similar to the YOLO Algorithm. Sometimes, during foggy conditions - obstacles are not clear to naked eyes, but Saliency map shows the obstacle in a distinct manner. The pre-trained model was used for the implementation of YOLO object detection algorithm. The dataset comprised of potential objects encountered as obstacles on roads especially while driving a vehicle. Sharp RGB image frames and the saliency map of the image frame were fused together. We have proposed our algorithm for this fusion process that takes into consideration the bounding boxes generated if the object is found in YOLO predicted by Saliency map and the average confidence scores for bounding boxes higher than a threshold in the saliency map in the region predicted by YOLO. Ultimately, union of all the bounding boxes is done to obtain the optimum result. The prototype of the solution has been shown in Fig. 3.

B. Optical Fog Sensor

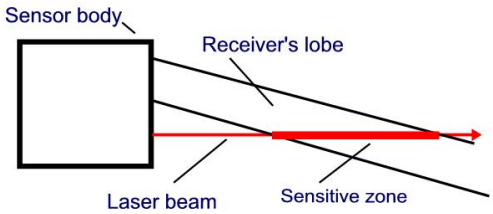


Fig 4 : Working of Optical Fog Sensor

The working of an optical fog sensor is based on backscatter technique. If the sensor gives positive output i.e. fog is sensed in the nearby environment, then the image sensor in the stereocamera is activated. Thus, the fog sensor acts as an activation signal in the overall VESY sensor. The sensor measures the amount of water particles in the air. Fig. 4 shows that a beam of laser light comes out of the sensor. There is a detector which is sensitive to incoming light in a narrow region that overlaps the transmitter beam. If there are fog particles in the overlap zone then the light will be scattered back. Light will reach the detector and send an activation signal to the image sensor. Little fog in the surrounding is enough to scatter the light and activate the image sensor. The image left and right sensors gives a clear image frame along with the depth map for the image. The image frame and the depth map are processed further before implementation of our proposed algorithm.

C. Depth Map

Using a stereo sensor, it is possible to infer the distance between two points with different depths in space having the same projection in the sensed image. Fig. 5 shows that coordinates in the target image have different positions. Based on a triangular relationship between the points, it is possible to infer the distance between the points. Fig. 5 contains equivalent triangles therefore the following result can be obtained by their equivalent equations :

$$\text{Disparity Value} = x - x' = \frac{Bf}{Z} \quad (1)$$

x' and x in equation (1) are the distance between points in image plane corresponding to the scene point 3D and their camera centre. f is the focal length of camera and distance between the two cameras has been represented by B. According to the equation (1) the depth of a coordinate in a scene has an inverse relation to the difference in distance of corresponding image coordinate and their camera centres. Using this data, we have derived the depth of all pixels in an image

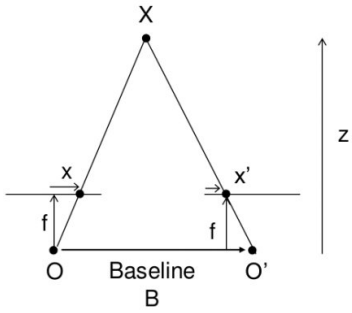


Figure 5 : Calculating Distance between object and viewer

D. Dehaze Algorithm

Dehaze algorithm was applied to each image frame obtained from the stereo sensor and the enhanced frame thus obtained was processed further to obtain the Saliency map. The algorithm is a three step process. Initially it estimates the intensity of the atmospheric light by searching in the sensed distorted image. Then the transmission map is calculated by describing a coarse map from a fine map. Finally, it will produce a clear image from the foggy image that we passed by using value of estimated intensity and transmission map. The results can be observed in Fig. 6 where the foggy image is enhanced by application of Dehaze algorithm. Initially, intensity of atmospheric light A is estimated from image I(x). We find the topmost 0.1% pixels according to their brightness in the dark channel. Pixel with highest intensity is chosen as the representative of atmospheric light. Transmission map t(x) is then estimated using using A and I(x). We first find a dark channel in regard to on a local area (Coarse Map) and then t(x) by :

$$t(x) = 1 - \text{defoggingParam} * \text{darkPixelFromCoarseMap} / \text{AtmosphericLightIntensity} \quad (2)$$

The defoggingParam in (2) is a value between 0 to 1. The higher value the lesser amount of fog would be kept for the distant objects.We can clarify the images by using the equation that follows:

$$K(x) = (I(x) - A) / \max(t(x), t(0)) + A \quad (3)$$

Equation (3) uses the following variables : K(x) is output, I(x) is input, t(x) is transmission map, A is atmospheric light and t(0) is set to a particular value to avoid dividing by zero.

Figure 6 : Result of Dehaze Algorithm to enhance Foggy Image

Figure 7 : Saliency Map for Fig. 5

E. Saliency Map

The enormous amount of information especially visual that is being sensed is so vast that various mechanisms are harnessed for choosing the most suitable part of the visual data under consideration and the rest of the part is often discarded. Reference [10] uses 2 state of the art static saliency mechanisms which are dependent on covariance based on region to process extra information, related to the depth that is available in RGB-D images. RGB-D image is an image and comprises of 4 channels. The first 3 channels of RGB-D form the general RGB image, whereas the last channel or the fourth one denotes a depth channel in alignment to the RGB component. We extend the CovSal saliency models [16] and implement them on RGB- D images. These models have been implemented on RGB image dataset which predicts saliency by accurately guessing the centre surrounding differences on the basis of 1st and 2nd order feature statistics. In our work, these models have been employed for estimating two different types of saliency maps, one from the general RGB image whereas the second one from the depth image. RGB and depth saliency maps are then combined thus giving a single saliency image for the given RGB-D as output. When compared with other state of art methods, the proposed technique of Region Covariance Matrix shows better versatility and accuracy for saliency detection of objects in foggy images. Fig. 7 shows the generated Saliency Map on one of our image frames. This result has been obtained by processing the image obtained after application of Dehaze algorithm.

F. YOLO Real Time Object Detection

The input image is split into S × S boxes by YOLO. The cell within which the centre of the object falls leads to the detection of that item. Each element in the grid estimates B number of bounding boxes and confidence values associated with every box. The confidence values shows how sure the hypotheses is that the bounding box contains that item. Confidence has been given a proper definition, as Pr(Object) ∗ IOU(Intersection Over Union). Confidence scores should be zero if no identifiable object exists in the box. Else we aim at getting the confidence value to be equivalent to the IOU between box that was predicted and the ground truth assumed. Every bounded box comprises of five predictions: x, y, w, h, along with the confidence. The mid point of the box is usually showcased by the (x,y) coordinates. (w,h) parameters are used as an alias for the image width and height. The IOU achieved between the box that was predicted and box representing ground truth gives the final confidence prediction. Each cell of

Layer (type)	Output Shape	Param #	Connected to
convolution2d_1 (Convolution2D)	(None, 16, 448, 448)	448	convolution2d_input_1[0][0]
leakyrelu_1 (LeakyReLU)	(None, 16, 448, 448)	0	convolution2d_1[0][0]
maxpooling2d_1 (MaxPooling2D)	(None, 16, 224, 224)	0	leakyrelu_1[0][0]
convolution2d_2 (Convolution2D)	(None, 32, 224, 224)	4640	maxpooling2d_1[0][0]
leakyrelu_2 (LeakyReLU)	(None, 32, 224, 224)	0	convolution2d_2[0][0]
maxpooling2d_2 (MaxPooling2D)	(None, 32, 112, 112)	0	leakyrelu_2[0][0]
convolution2d_3 (Convolution2D)	(None, 64, 112, 112)	18496	maxpooling2d_2[0][0]
leakyrelu_3 (LeakyReLU)	(None, 64, 112, 112)	0	convolution2d_3[0][0]
maxpooling2d_3 (MaxPooling2D)	(None, 64, 56, 56)	0	leakyrelu_3[0][0]
convolution2d_4 (Convolution2D)	(None, 128, 56, 56)	73856	maxpooling2d_3[0][0]
leakyrelu_4 (LeakyReLU)	(None, 128, 56, 56)	0	convolution2d_4[0][0]
maxpooling2d_4 (MaxPooling2D)	(None, 128, 28, 28)	0	leakyrelu_4[0][0]
convolution2d_5 (Convolution2D)	(None, 256, 28, 28)	295168	maxpooling2d_4[0][0]
leakyrelu_5 (LeakyReLU)	(None, 256, 28, 28)	0	convolution2d_5[0][0]
maxpooling2d_5 (MaxPooling2D)	(None, 256, 14, 14)	0	leakyrelu_5[0][0]
convolution2d_6 (Convolution2D)	(None, 512, 14, 14)	1180160	maxpooling2d_5[0][0]
leakyrelu_6 (LeakyReLU)	(None, 512, 14, 14)	0	convolution2d_6[0][0]
maxpooling2d_6 (MaxPooling2D)	(None, 512, 7, 7)	0	leakyrelu_6[0][0]
convolution2d_7 (Convolution2D)	(None, 1024, 7, 7)	4719616	maxpooling2d_6[0][0]
leakyrelu_7 (LeakyReLU)	(None, 1024, 7, 7)	0	convolution2d_7[0][0]
convolution2d_8 (Convolution2D)	(None, 1024, 7, 7)	9438208	leakyrelu_7[0][0]
leakyrelu_8 (LeakyReLU)	(None, 1024, 7, 7)	0	convolution2d_8[0][0]
convolution2d_9 (Convolution2D)	(None, 1024, 7, 7)	9438208	leakyrelu_8[0][0]
leakyrelu_9 (LeakyReLU)	(None, 1024, 7, 7)	0	convolution2d_9[0][0]
flatten_1 (Flatten)	(None, 50176)	0	leakyrelu_9[0][0]
dense_1 (Dense)	(None, 256)	12845312	flatten_1[0][0]
dense_2 (Dense)	(None, 4096)	1052672	dense_1[0][0]
leakyrelu_10 (LeakyReLU)	(None, 4096)	0	dense_2[0][0]
dense_3 (Dense)	(None, 1470)	6022590	leakyrelu_10[0][0]

Total params: 45,089,374
Trainable params: 45,089,374
Non-trainable params: 0

Figure 8 : Layers in YOLO Implementation

the grid predicts conditional probabilities of class, Pr(Class i | Object). The probabilities have been conditioned on the cell in grid which contains the desired object. One part of probabilities of class are predicted in each cell of grid as defined, nevertheless of the count of boxes. During test time multiplication of conditional class probabilities is done and confidence predictions of every box was made individually.

$$\Pr(\text{Classi}|\text{Object})*\Pr(\text{Object})*\text{IOU} = \Pr(\text{Classi})*\text{IOU} \quad (4)$$

Equation (4) gives the specific confidence values for each class for every box. Probabilities of those specific classes are shown inside the box and depicts how nicely the box positions are predicted for the object under consideration. The neural network has 9 convolutional layers. After these layers, 3 fully connected layers are added to make it faster. YOLO is optimised as it predicts many bounding boxes for every cell in grid. One bounding box predictor is given the responsibility for each object at training time. The algorithm assigns one predictor the task for detecting an object. The predictor does so by finding out the prediction which has the highest current intersection over union with the assumed ground truth. Specialisation is done between the bounding box predictors because of this. Overall recall is improved as performance of every predictor is enhanced for predicting certain sizes or

aspect ratios. Fig. 8 shows the 9 convolutional layers followed by 3 fully connected layers used in our implementation. Implementation of YOLO detection algorithm on foggy images is depicted in Fig. 1. It is found that only some nearby and clear objects are detected by the object detection algorithm. Bounding boxes were not formed for the unclear objects. The other objects were clearly evident in the saliency map of the image and hence the saliency map can be used to form bounding boxes for the remaining undetected objects.

G. VESY Algorithm

The Saliency map generated is fused with results of YOLO algorithm. The following terminology has been used :
P(Yolo|Saliency) : Probability that object is found by YOLO in the region predicted by Saliency Map.
P(Saliency|Yolo) : Mean of Pixel Values of the Saliency Map in the region predicted by YOLO.

Proposed Algorithm

Variables Used : μ_s (Threshold for Saliency Map), u_y (Threshold for the confidence value of YOLO to mark bounding box), $A_{s[nxm]}$ (Auxiliary Array of the Dimensions of the Saliency Map), $S_{[nxm]}$ (Saliency Matrix), Y (YOLO produced Matrix), Y' (YOLO produced Matrix at a low threshold), R_s [lt, rt, lb, rb] (Represent the corners of square sub matrices), μ_{ys} (Threshold for finding Bounding Boxes for objects found in YOLO in the region predicted by Saliency Map), μ_{sy} (Threshold for finding Bounding Box for objects with pixel values >= μ_s in the Saliency Map in the region predicted by YOLO).

Step 1 : Create Saliency Matrix S with intensities of white regions for each pixel of the sensed image.

Step 2 : Run YOLO and generate matrix Y at threshold u_y and Y' (at a low threshold) which comprises of 5 predictions: x, y, w, h, and confidence values.

Step 3 : P(Yolo|Saliency) is computed:

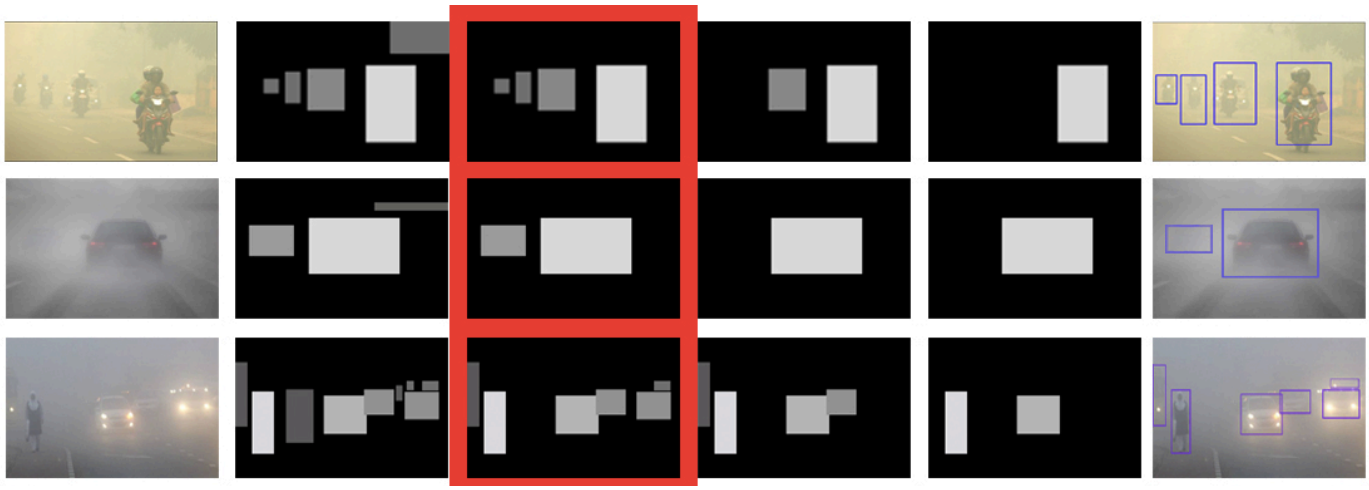
```
1   for i = 0 to n-1
2       for j = 0 to m-1
3           if Sij >= μs
4               Aij=1
5   find the max size square sub-matrices with
    Aij=1 and store their dimensions in Rs
6   for i = 0 to Rs.length - 1
7       a, b, c, d = Rs[i].lt, Rs[i].rt, Rs[i].lb, Rs[i].rb
8       sal_avg = Mean (Sij bounded by a, b, c, d)
9       y_avg  = Find Average Confidence Value in
                 Y'  for Box bounded by a, b, c, d
10      if (mean( sal_avg, y_avg) > μys)
11          mark_box  (a, b, c, d)
```

Step 4 : P(Saliency|Yolo) is computed

```
1   for i = 0 to Y.length - 1

2       a, b, c, d = Y[i].lt, Y[i].rt, Y[i].lb, Y[i].rb
3       y_avg  = Y[i].confidence
4       sal_avg = Mean (Sij bounded by a, b, c, d)
5       if (mean( sal_avg, y_avg) > μsy)
6           mark_box  (a, b, c, d)
```

Step 5 : The required output is Union of the Bounding Boxes generated by P(Yolo|Saliency) & P(Saliency|Yolo) i.e. *(P(Saliency|Yolo) U P(Yolo|Saliency))*. The above computation provides all the Bounding Box that cross the Bounding Function specified in their respective algorithms. Bounding Function in both the computations is chosen as the Threshold Values (μ_{sy} & μ_{ys}). The bounding boxes that cross the threshold our considered a part of the desired solution.



(a) (b) (c) (d) (e) (f)

Figure 9 : Results obtained after implementation of our algorithm

IV. RESULT AND DISCUSSION

The value of threshold μ_{sy} is kept fixed and results are computed by varying the value of threshold μ_{ys} for various images. The results have been computed by keeping the value of threshold μ_{ys} as 0.60, 0.70, 0.80 and 0.90. The algorithm proposed in the previous section was tested on several images and the results for three images have been showcased in Figure 9. Fig. 9 (a) shows the original image used for the implementation of the algorithm. Fig. 9 (b) shows the bounding boxes at $\mu_{ys} = 0.60$, Fig. 9 (c) shows the bounding boxes at $\mu_{ys} = 0.70$, Fig. 9 (d) shows the bounding boxes at $\mu_{ys} = 0.80$, Fig. 9 (e) shows the bounding boxes at $\mu_{ys} = 0.90$ and Fig. 9 (f) shows the final output on the input image.

It is observed that when the threshold is kept at a low value, all the objects are detected and their bounding boxes are generated even if their presence is not distinct in either YOLO object detection or the saliency map. $\mu_{ys} = 0.70$ gives an optimum result yielding bounding boxes for all the objects in all the three images. These bounding boxes are a result of union of the bounding boxes generated by YOLO object detection and saliency map. Thus, $\mu_{ys} = 0.70$ is a suitable value for the threshold. The result also shows that not all objects are able to cross the high threshold and thus only few distinct bounding boxes are generated in such cases. Therefore, $\mu_{ys} = 0.80$ is not a suitable value for the threshold. Similarly, $\mu_{ys} = 0.90$ has even a higher value of threshold and thus yields poor results. Therefore, $\mu_{ys} = 0.90$ is not a suitable value for the threshold.

V. CONCLUSION

The union of the bounding boxes generated by YOLO algorithm and the Saliency map is achieved at the desired threshold value using the proposed algorithm. The final output detects all the objects in a foggy image frame. The proposed VESY(Visibility Enhancement Saliency YOLO) algorithm is capable of detecting additional objects that the YOLO algorithm was unable to detect. Also, distance can be calculated via the depth map and can be displayed in addition to the bounding boxes around the objects for application in Real-Time Systems.

REFERENCES

1. Zhiming Tan, Xianghui Bai, Bingrong Wang and Akihiro Higashi, “Fast Single-image Defogging” FUJITSU Sci. Tech. J., vol. 50, No. 1, January 2014.
2. Lu W., Sun X., Li C. , “A New Method of Object Saliency Detection in Foggy Images”, Image and Graphics 8th International Conference, ICIG 2015, Tianjin, China, August, 2015, vol. 9217, pp.206-217.
3. Olga Krutikova, Aleksandrs Sisojevs and Mihails Kovalovs, “Creation of a Depth Map from Stereo Images of Faces for 3D Model Reconstruction”, Procedia Computer Science, 2017, vol. 104, pp. 452-459.
4. P. R. Induchoodan, M. J. Josemartin and P. R. Geetharanjin, "Depth recovery from stereo images," 2014 International Conference on Contemporary Computing and Informatics (IC3I), Mysore, 2014, pp. 745-750.
5. H. Song, Z. Liu, H. Du, G. Sun, O. Le Meur and T. Ren, "Depth-Aware Salient Object Detection and Segmentation via Multiscale Discriminative Saliency Fusion and Bootstrap Learning," in IEEE Transactions on Image Processing, vol. 26, no. 9, pp. 4204-4216, Sept. 2017.
6. Guo, Jingfan & Ren, Tongwei & Bei, Jia & Zhu, Yujin. (2015), “Salient object detection in RGB-D image based on saliency fusion and propagation”, The 7th International Conference, August 2015.
7. Redmon Joseph, Divvala Santosh, Girshick Ross, Farhadi Ali, “You Only Look Once: Unified, Real-Time Object Detection”, eprint arXiv: 1506.02640, June, 2015.
8. Fan Guo, Jin Tang, Zixing Cai, “Fusion Strategy for Single Image Dehazing”, International Journal of Digital Content Technology and its Applications(JDCTA), vol. 7, January 2013.
9. S. Ansia, A.L. Aswathy, “Single Image Haze Removal Using White Balancing and Saliency Map”, Procedia Computer Science, 2015, vol. 46, pp. 12-19.
10. Erkut Erdem, “A Region Covariances-based Visual Attention Model for RGB-D Images”, International Journal of Intelligent Systems and Applications in Engineering, vol. 4, pp. 128-134, December, 2016.
11. Wang ST., Zhou Z., Qu HB., Li B. (2017), “Visual Saliency Detection for RGB-D Images with Generative Model”, Computer Vision – ACCV 2016. ACCV 2016, Lecture Notes in Computer Science, vol 10115. Springer, Cham.
12. R. Cong, J. Lei, C. Zhang, Q. Huang, X. Cao and C. Hou, "Saliency Detection for Stereoscopic Images Based on Depth Confidence Analysis and Multiple Cues Fusion," in IEEE Signal Processing Letters, vol. 23, no. 6, pp. 819-823, June 2016.
13. Jiawei Xu, Shigang Yue, “Fusion of Saliency Maps for Visual Attention Selection in Dynamic Scenes”, International Journal of Advanced Research in Artificial Intelligence(IJARAI),2013, vol. 2.
14. L. Qu, S. He, J. Zhang, J. Tian, Y. Tang and Q. Yang, "RGBD Salient Object Detection via Deep Fusion," in IEEE Transactions on Image Processing, vol. 26, no. 5, pp. 2274-2285, May 2017.
15. Juan Du, “Understanding of Object Detection Based on CNN Family and YOLO”, Journal of Physics: Conference Series, 2018, vol. 1004.
16. E. Erdem, and A. Erdem , “Visual saliency estimation by nonlinearly integrating features using region covariances”, Journal of Vision, vol. 13, pp. 1-20.