International Journal of Scientific Research in Science and Technology
Available online at : www.ijsrst.com
Print ISSN: 2395-6011 | Online ISSN: 2395-602X
doi : https://doi.org/10.32628/IJSRST

Advanced Multi-spectral Object detection using Night Vision Surveillance

[1] T. Veena, [2] Dokku Sesha Joshna, [3] Mohammad Sattar Basha, [4] Palagani Bhagyasri, [5] Chinnam Sairajeev
[1]Assistant Professor, [2,3,4,5] UG Students
Department of CSE-AI & ML, Sri Vasavi Institute of Engineering & Technology, Nandamuru, Andhra Pradesh, India

ARTICLEINFO

Article History:

Accepted: 15 April 2024
Published: 27 April 2024

Publication Issue :
Volume 11, Issue 2
March-April-2024

Page Number :
854-861

ABSTRACT

Surveillance has become an important task in recent time mainly due to the increasing of crime rates. The existing research on surveillance for day time has achieved better performance by detecting and tracking objects using deep learning algorithms. However, it is difficult to achieve the same performance for night vision mainly due to low illumination and/or bad weather situation. One of the important tasks in surveillance is object detection which results in both class and location of the detected object with clear boundary of the objects from the image. We propose an efficient object detection module using fusion of thermal and visible images. Fusion module consists of encoder-decoder network in which encoder uses depthwise convolution to extracts efficient features from the given thermal and visible images. Then after, fused image is reconstructed using convolutional layers and final map is utilized in object detection algorithm (i.e., mask RCNN). The proposed method shows the effectiveness of utilization of pre-processing module i.e., fusion in object detection algorithm. Here, it is observed that the proposed method performs better for night vision when images are trained carefully with various features. Moreover, proposed method performs better on real time night vision images having no illumination condition.

Keywords : Depth Wise Convolution, Encoder-Decoder Network, Image Fusion, Night Vision Thermal Images, Object Detection.

I. INTRODUCTION

In the recent scenario of increased security and surveillance, more robust and sophisticated surveillance systems are demanded. Among many types of deep learning models, deep convolutional neural network (DCNN) is a powerful approach for low to high level feature learning. The main aim behind the use of DCNN is to extract features effectively from the data captured in low or no illumination situation during night time. Recently, thermal infrared camera is widely used to detect object in low illumination situation. The visible cameras have ability to capture images under natural/ artificial illumination conditions only. Hence, very limited visual information are captured in night vision and that makes difficult to perform surveillance in night time using visual sensors only. Moreover, thermal images contain higher information of objects which have high temperature. However, for the objects having low temperature, it provides poor information. On the other hand, visual imaging contains the high visual context of the particular object.

Recently, rapidly increasing trend of deep learning algorithms outperform on the application of object detection from thermal images. Nevertheless huge improvements have been manifested in recent time to develope a method for efficient detection that is effective for practical applications still remains a challenging problem. It is observed that most existing object detection methods are sensitive to changes of light, climate and impediments due to perform training operation on visual information only. To overcome the previously mentioned limitations for night time object detection, many research problems have been targeted on the development of multi-spectral object detection solutions for facilitating robust target detection.

Surveillance has become an important task in recent time mainly due to the increasing of crime rates. The existing research on surveillance for day time has achieved better performance by detecting and tracking objects using deep learning algorithms. However, it is difficult to achieve the same performance for night vision mainly due to low illumination and/or bad weather situation. One of the important tasks in surveillance is object detection which results in both class and location of the detected object with clear boundary of the objects from the image.

II.RELATED WORK

Title: Thermal Cameras and Applications: A Survey

Thermal cameras are passive sensors that capture the infrared radiation emitted by all objects with a temperature above absolute zero. This type of camera was originally developed as a surveillance and night vision tool for the military, but recently the price has dropped, significantly opening up a broader field of applications. Deploying this type of sensor in vision systems eliminates the illumination problems of normal greyscale and RGB cameras. This survey provides an overview of the current applications of thermal cameras. Applications include animals, agriculture, buildings, gas detection, industrial, and military applications, as well as detection, tracking, and recognition of humans. Moreover, this survey describes the nature of thermal radiation and the technology of thermal cameras

Title: Multispectral Deep Neural Networks for Pedestrian Detection

Multispectral pedestrian detection is essential for around-the-clock applications, e.g., surveillance and autonomous driving. We deeply analyze Faster R-CNN for multispectral pedestrian detection task and then model it into a convolutional network (ConvNet) fusion problem. Further, we discover that ConvNet-based pedestrian detectors trained by color or thermal images separately provide complementary information in discriminating human instances. Thus there is a large potential to improve pedestrian detection by using color and thermal images in DNNs simultaneously. We carefully design four ConvNet fusion architectures that integrate two-branch ConvNets on different DNNs stages, all of which yield better performance compared with the baseline detector. Our experimental results on KAIST pedestrian benchmark show that the Halfway Fusion model that performs fusion on the middle-level convolutional features outperforms the baseline method by 11% and yields a missing rate 3.5% lower than the other proposed architectures

Title: Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks

State-of-the-art object detection networks depend on region proposal algorithms to hypothesize object locations. Advances like SPPnet [1] and Fast R-CNN [2] have reduced the running time of these detection networks, exposing region proposal computation as a bottleneck. In this work, we introduce a Region Proposal Network(RPN) that shares full-image convolutional features with the detection network, thus enabling nearly cost-free region proposals. An RPN is a fully convolutional network that simultaneously predicts object bounds and objectness scores at each position. The RPN is trained end-to-end to generate high-quality region proposals, which are used by Fast R-CNN for detection. We further merge RPN and Fast R-CNN into a single network by sharing their convolutional features-using the recently popular terminology of neural networks with 'attention' mechanisms, the RPN component tells the unified network where to look. For the very deep VGG-16 model [3], our detection system has a frame rate of 5 fps (including all steps) on a GPU, while achieving state-of-the-art object detection accuracy on PASCAL VOC 2007, 2012, and MS COCO datasets with only 300 proposals per image. In ILSVRC and COCO 2015 competitions, Faster R-CNN and RPN are the foundations of the 1st-place winning entries in several tracks. Code has been made publicly available.

III.PROPOSED SYSTEM

We propose an efficient object detection module using fusion of thermal and visible images. Fusion module consists of encoder-decoder network in which encoder uses depthwise convolution to extracts efficient features from the given thermal and visible images. Then after, fused image is reconstructed using convolutional layers and final map is utilized in object detection algorithm (i.e., mask RCNN). The proposed method shows the effectiveness of utilization of pre-processing module i.e., fusion in object detection algorithm.

Advantages:

- Observed that the proposed method performs better for night vision when images are trained carefully with various features.
- Moreover, proposed method performs better on real time night vision images having no illumination condition.

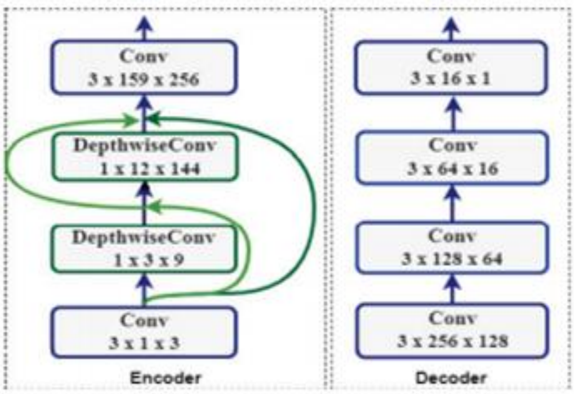

Fig 1: Block Diagram

3.1 MODULES DESCRIPTION

RCNN: A Fast R-CNN network takes as input an entire image and a set of object proposals. The network first processes the whole image with several convolutional (conv) and max pooling layers to produce a conv feature map. Then, for each object proposal a region of interest (RoI) pooling layer extracts a fixed-length feature vector from the feature map. To achieve real time speeds, Faster RCNN uses "REGION PROPOSAL NETWORKS".

Pandas: pandas is an open source, BSD-licensed library providing high-performance, easy-to-use data structures and data analysis tools for the Python programming language.

Numpy: NumPy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays. It is the fundamental package for scientific computing with Python.

MatPlotLib: matplotlib.Pyplot is a plotting library used for 2D graphics in python programming language. It can be used in python scripts, shell, web application servers and other graphical user interface toolkits

TensorFlow: TensorFlow is an end-to-end open source platform for machine learning. It has a comprehensive, flexible ecosystem of tools, libraries and community resources that lets researchers push the state-of-the-art in ML and developers easily build and deploy ML powered applications.

IV.RESULTS AND DISCUSSION

```
import cv2
import matplotlib.pyplot as plt
%matplotlib inline
```

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

Fig 2. Results Screenshot

```
# verify CUDA
!/usr/local/cuda/bin/nvcc --version
```

```
nvcc: NVIDIA (R) Cuda compiler driver
Copyright (c) 2005-2020 NVIDIA Corporation
Built on Wed_Jul_22_19:09:09_PDT_2020
Cuda compilation tools, release 11.0, V11.0.221
Build cuda_11.0_bu.TC445_37.28845127_0
```

```
# make darknet (builds darknet so that you can then use the darknet executable file to run or train object detectors)
!make
```

Fig 3. Results Screenshot

```
g++ -std=c++11 -std=c++11 -Iinclude/ -I3rdparty/stb/include -DOPENCV `pkg-config --cflags opencv4 2> /dev/null || pkg-config --cflags opencv
./src/image_opencv.cpp: In function ‘void draw_detections_cv_v3(void**, detection*, int, float, char**, image**, int, int)’:
./src/image_opencv.cpp:926:23: warning: variable ‘rgb’ set but not used [-Wunused-but-set-variable]
                 float rgb[3];
                       ^~~
./src/image_opencv.cpp: In function ‘void draw_train_loss(char*, void**, int, float, float, int, int, float, int, char*, float, int, int, do
./src/image_opencv.cpp:1127:13: warning: this ‘if’ clause does not guard... [-Wmisleading-indentation]
             if (iteration_old == 0)
             ^~
./src/image_opencv.cpp:1130:10: note: ...this statement, but the latter is misleadingly indented as if it were guarded by the ‘if’
          if (iteration_old != 0){
          ^~
./src/image_opencv.cpp: In function ‘void cv_draw_object(image, float*, int, int, int*, float*, int*, int, char**)’:
./src/image_opencv.cpp:1424:14: warning: unused variable ‘buff’ [-Wunused-variable]
         char buff[100];
              ^~~~
./src/image_opencv.cpp:1400:9: warning: unused variable ‘it_tb_res’ [-Wunused-variable]
     int it_tb_res = cv::createTrackbar(it_trackbar_name, window_name, &it_trackbar_value, 1000);
         ^~~~~~~~~
./src/image_opencv.cpp:1404:9: warning: unused variable ‘lr_tb_res’ [-Wunused-variable]
     int lr_tb_res = cv::createTrackbar(lr_trackbar_name, window_name, &lr_trackbar_value, 20);
         ^~~~~~~~~
./src/image_opencv.cpp:1408:9: warning: unused variable ‘cl_tb_res’ [-Wunused-variable]
     int cl_tb_res = cv::createTrackbar(cl_trackbar_name, window_name, &cl_trackbar_value, classes-1);
         ^~~~~~~~~
./src/image_opencv.cpp:1411:9: warning: unused variable ‘bo_tb_res’ [-Wunused-variable]
     int bo_tb_res = cv::createTrackbar(bo_trackbar_name, window_name, boxonly, 1);
         ^~~~~~~~~
g++ -std=c++11 -std=c++11 -Iinclude/ -I3rdparty/stb/include -DOPENCV `pkg-config --cflags opencv4 2> /dev/null || pkg-config --cflags opencv
In file included from ./src/http_stream.cpp:580:0:
```

Fig 4. Results Screenshot

```
nvcc -gencode arch=compute_35,code=sm_35 -gencode arch=compute_50,code=[sm_50,compute_50] -gencode arch=compute_52,code=[sm_52,compute_52] -
nvcc warning : The 'compute_35', 'compute_37', 'compute_50', 'sm_35', 'sm_37' and 'sm_50' architectures are deprecated, and may be removed i
nvcc -gencode arch=compute_35,code=sm_35 -gencode arch=compute_50,code=[sm_50,compute_50] -gencode arch=compute_52,code=[sm_52,compute_52] -
nvcc warning : The 'compute_35', 'compute_37', 'compute_50', 'sm_35', 'sm_37' and 'sm_50' architectures are deprecated, and may be removed i
nvcc -gencode arch=compute_35,code=sm_35 -gencode arch=compute_50,code=[sm_50,compute_50] -gencode arch=compute_52,code=[sm_52,compute_52] -
nvcc warning : The 'compute_35', 'compute_37', 'compute_50', 'sm_35', 'sm_37' and 'sm_50' architectures are deprecated, and may be removed i
nvcc -gencode arch=compute_35,code=sm_35 -gencode arch=compute_50,code=[sm_50,compute_50] -gencode arch=compute_52,code=[sm_52,compute_52] -
nvcc warning : The 'compute_35', 'compute_37', 'compute_50', 'sm_35', 'sm_37' and 'sm_50' architectures are deprecated, and may be removed i
./src/network_kernels.cu(364): warning: variable "l" was declared but never referenced

./src/network_kernels.cu(364): warning: variable "l" was declared but never referenced

./src/network_kernels.cu(364): warning: variable "l" was declared but never referenced

./src/network_kernels.cu(364): warning: variable "l" was declared but never referenced

./src/network_kernels.cu(364): warning: variable "l" was declared but never referenced

./src/network_kernels.cu: In function ‘float train_network_datum_gpu(network, float*, float*)’:
./src/network_kernels.cu:364:7: warning: variable ‘l’ set but not used [-Wunused-but-set-variable]
     layer l = net.layers[net.n - 1];
       ^
nvcc -gencode arch=compute_35,code=sm_35 -gencode arch=compute_50,code=[sm_50,compute_50] -gencode arch=compute_52,code=[sm_52,compute_52] -
nvcc warning : The 'compute_35', 'compute_37', 'compute_50', 'sm_35', 'sm_37' and 'sm_50' architectures are deprecated, and may be removed i
g++ -std=c++11 -std=c++11 -Iinclude/ -I3rdparty/stb/include -DOPENCV `pkg-config --cflags opencv4 2> /dev/null || pkg-config --cflags opencv
```

Fig 5. Results Screenshot

```
# define helper functions
def imShow(path):
  import cv2
  import matplotlib.pyplot as plt
  %matplotlib inline

  image = cv2.imread(path)
  height, width = image.shape[:2]
  resized_image = cv2.resize(image,(3*width, 3*height), interpolation = cv2.INTER_CUBIC)

  fig = plt.gcf()
  fig.set_size_inches(18, 10)
  plt.axis("off")
  plt.imshow(cv2.cvtColor(resized_image, cv2.COLOR_BGR2RGB))
  plt.show()

# use this to upload files
def upload():
  from google.colab import files
  uploaded = files.upload()
  for name, data in uploaded.items():
    with open(name, 'wb') as f:
      f.write(data)
      print ('saved file', name)

# use this to download a file
def download(path):
  from google.colab import files
  files.download(path)
```

Fig 6. Results Screenshot

```
# run darknet detection on test images
!./darknet detector test cfg/coco.data cfg/yolov4.cfg yolov4.weights data/person.jpg
```

```
 124 conv    256       1 x 1/ 1     38 x  38 x 512 ->   38 x  38 x 256 0.379 BF
 125 conv    512       3 x 3/ 1     38 x  38 x 256 ->   38 x  38 x 512 3.407 BF
 126 conv    256       1 x 1/ 1     38 x  38 x 512 ->   38 x  38 x 256 0.379 BF
 127 conv    128       1 x 1/ 1     38 x  38 x 256 ->   38 x  38 x 128 0.095 BF
 128 upsample                 2x    38 x  38 x 128 ->   76 x  76 x 128
 129 route  54                                     ->   76 x  76 x 256
 130 conv    128       1 x 1/ 1     76 x  76 x 256 ->   76 x  76 x 128 0.379 BF
 131 route  130 128                                ->   76 x  76 x 256
 132 conv    128       1 x 1/ 1     76 x  76 x 256 ->   76 x  76 x 128 0.379 BF
 133 conv    256       3 x 3/ 1     76 x  76 x 128 ->   76 x  76 x 256 3.407 BF
 134 conv    128       1 x 1/ 1     76 x  76 x 256 ->   76 x  76 x 128 0.379 BF
 135 conv    256       3 x 3/ 1     76 x  76 x 128 ->   76 x  76 x 256 3.407 BF
 136 conv    128       1 x 1/ 1     76 x  76 x 256 ->   76 x  76 x 128 0.379 BF
 137 conv    256       3 x 3/ 1     76 x  76 x 128 ->   76 x  76 x 256 3.407 BF
 138 conv    255       1 x 1/ 1     76 x  76 x 256 ->   76 x  76 x 255 0.754 BF
 139 yolo
[yolo] params: iou loss: ciou (4), iou_norm: 0.07, obj_norm: 1.00, cls_norm: 1.00, delta_norm: 1.00, scale_x_y: 1.20
nms_kind: greedynms (1), beta = 0.600000
```

Fig 7. Results Screenshot

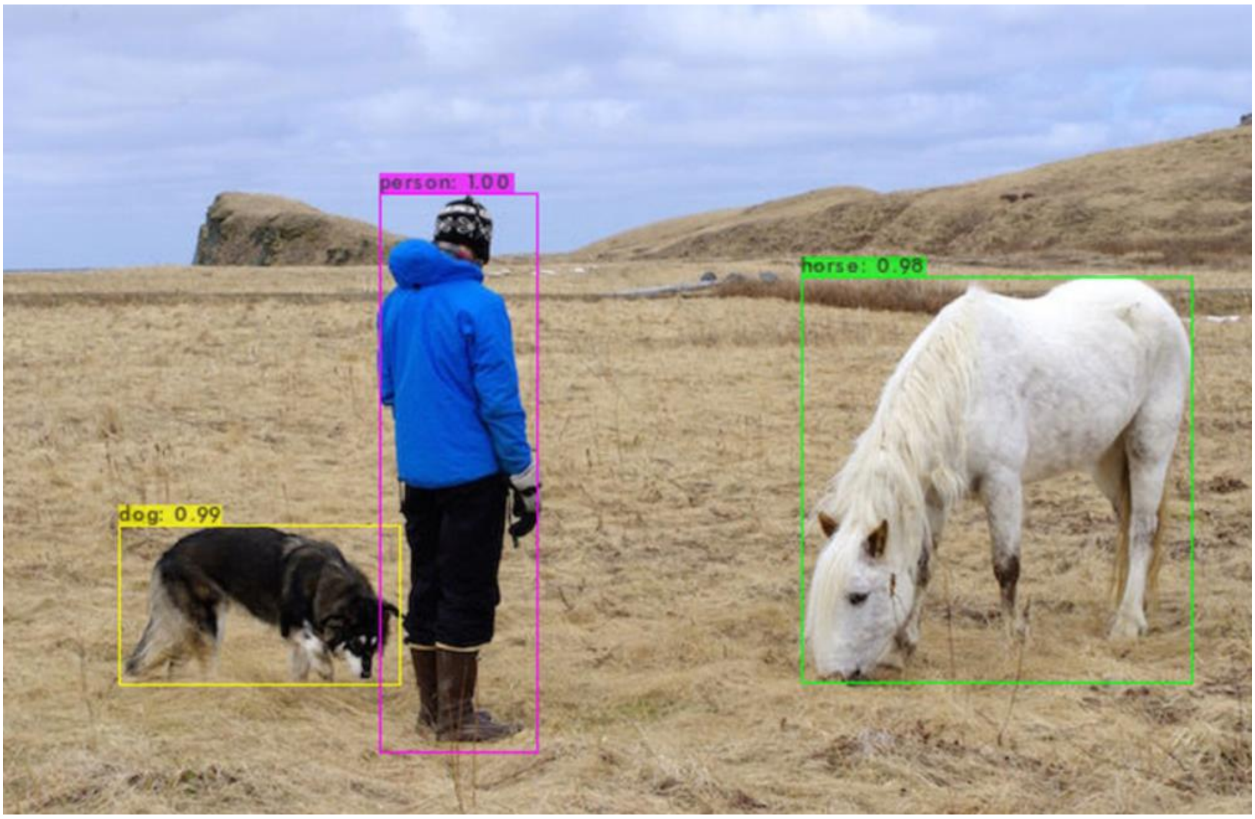

Fig 8. Results Screenshot

```
# try out the upload helper function! (I uploaded an image called highway.jpg, upload whatever you want!)
%cd ..
upload()
%cd darknet
```

```
/content
Choose Files  Streetb4.png
• Streetb4.png(image/png) - 152193 bytes, last modified: 3/20/2018 - 100% done
Saving Streetb4.png to Streetb4.png
saved file Streetb4.png
/content/darknet
```

Fig 9. Results Screenshot

Fig 10. Results Screenshot

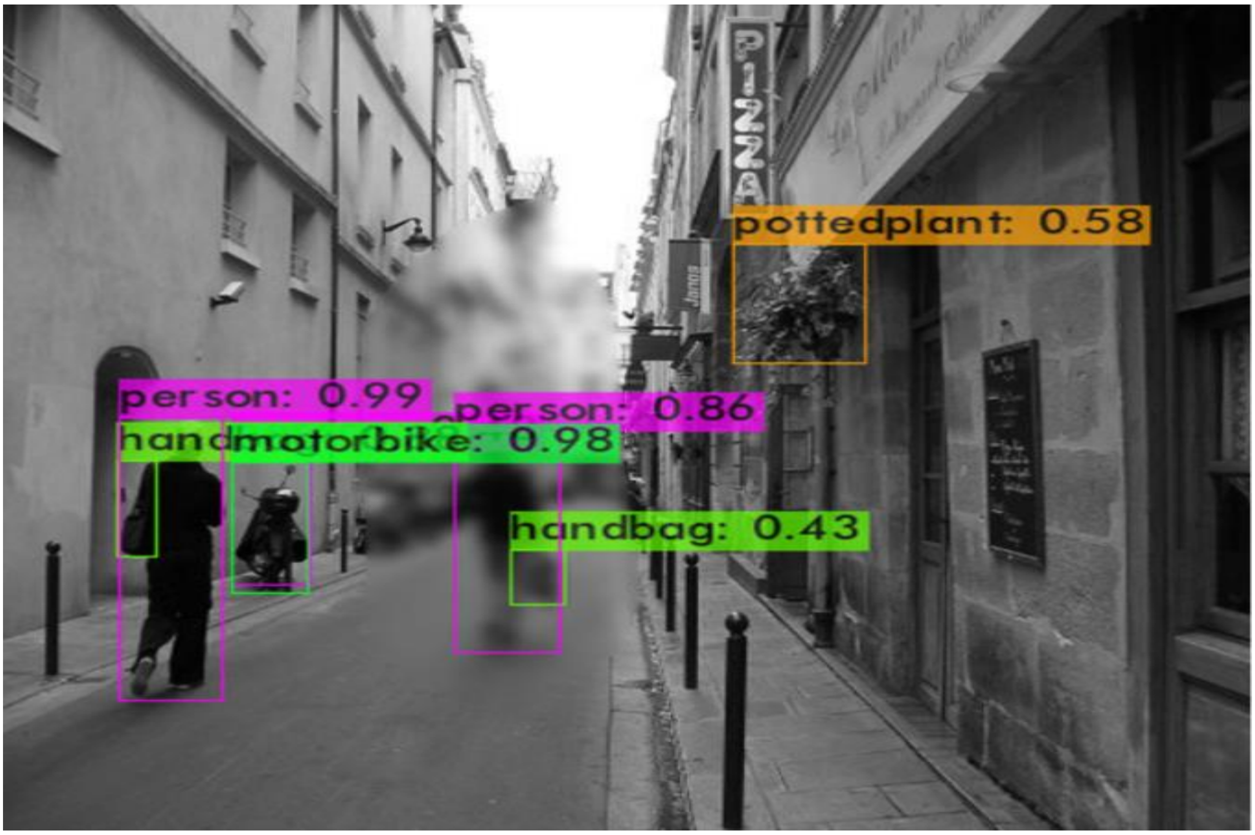

Fig 11. Results Screenshot

V.CONCLUSION

In this paper, we have presented an approach for object detection using thermal and visible images for night vision surveillance. The proposed network includes fusion and MRCNN module in which fusion module uses an encoder and decoder module with a depthwise convolution to extract salient features from the given input images. Then after, fused image is utilized to detect objects efficiently.

VI.FUTURE WORK

The experiments have been conducted on various datasets and missing rate is also calculated to verify the performance of the proposed method on real time night vision images. It shows that the proposed object detection method outperforms than the other state-of the-art existing methods.

II. REFERENCES

[1]. R. Gade and T. B. Moeslund, "Thermal cameras and applications: a survey," Machine vision and applications, vol. 25, no. 1, pp. 245–262, 2014.

[2]. J. Liu, S. Zhang, S. Wang, and D. N. Metaxas, "Multispectral deep neural networks for pedestrian detection," arXiv preprint arXiv:1611.02644, 2016.

[3]. J. Ma, Y. Ma, and C. Li, "Infrared and visible image fusion methods and applications: a survey," Information Fusion, vol. 45, pp. 153–178, 2019.

[4]. D. P. Bavirisetti and R. Dhuli, "Two-scale image fusion of visible and infrared images using saliency detection," Infrared Physics & Technology, vol. 76, pp. 52–64, 2016.

[5]. R. Gao, S. A. Vorobyov, and H. Zhao, "Image fusion with cosparse analysis operator," IEEE Signal Processing Letters, vol. 24, no. 7, pp. 943–947, 2017.

[6]. H. Li and X. Wu, "Densefuse: A fusion approach to infrared and visible images," IEEE Transactions on Image Processing, pp. 1–10, 2019.

[7]. S. Rajkumar, Mouli, and Chandra, "Infrared and visible image fusion using entropy and neuro-fuzzy concepts," in ICT and Critical Infrastructure: Proceedings of the 48th Annual Convention of Computer Society of India-Vol I. Springer, 2014, pp. 93–100.

[8]. J. Zhao, Y. Chen, H. Feng, Z. Xu, and Q. Li, "Infrared image enhancement through saliency feature analysis based on multi-scale decomposition," Infrared Physics & Technology, vol. 62, pp. 86–93, 2014.

[9]. Y. Liu, S. Liu, and Z. Wang, "A general framework for image fusion based on multi-scale transform and sparse representation," Information Fusion, vol. 24, pp. 147– 164, 2015.

[10].K. R. Prabhakar, V. S. Srikar, and R. V. Babu, "Deepfuse: A deep unsupervised approach for exposure fusion with extreme exposure image pairs," in 2017 IEEE International Conference on Computer Vision (ICCV). IEEE, 2017, pp. 4724–4732.

[11].G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2017, pp. 2261–2269.

[12].T. T. Zin, H. Takahashi, T. Toriu, and H. Hama, "Fusion of infrared and visible images for robust person detection," Image fusion, pp. 239–264, 2011.

[13].J. H. Kim, G. Batchuluun, and K. R. Park, "Pedestrian detection based on faster r-cnn in nighttime by fusing deep convolutional features of successive images," Expert Systems with Applications, vol. 114, pp. 15–33, 2018.

[14].D. Guan, Y. Cao, J. Yang, Y. Cao, and M. Y. Yang, "Fusion of multispectral data through illumination-aware deep neural networks for pedestrian detection," Information Fusion, vol. 50, pp. 148–157, 2019.