International Journal of Computer Trends and Technology
ISSN: 2231 – 2803 / https://doi.org/10.14445/22312803/IJCTT-V71I2P113

Volume 71 Issue 2, 87-92, February 2023

Original Article

Object Detection for Night Vision using Deep Learning Algorithms

Dipali Bhabad[1], Surabhi Kadam[2], Tejal Malode[3], Girija Shinde[4], Dipak Bage[5]

[1,2,3,4,5]*Department of Computer Engineering, K.K. Wagh Institute of Engineering Education and Research, Nashik Maharashtra, India.*

Received: 12 January 2023 Revised: 13 February 2023 Accepted: 23 February 2023 Published: 28 February 2023

Abstract *- Abnormal activity detection plays a very important role in surveillance applications. The existing research on surveillance for daytime 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. Deep learning is a powerful machine learning technique in which the object detector automatically learns image features required for detection tasks. It is required to generate a model that detects objects under low illumination. The approach is to use thermal infrared images and detect external objects, if any, and classify whether it is human or animal in an isolation area. With the rapid growth of deep learning, more efficient techniques will be implemented to solve the problems of object detection using neural networks and deep learning.*

Keywords *- Object Detection, RetinaNet, SSD, Thermal Infrared Images, YOLO.*

1. Introduction

Object detection is an important aspect of computer vision technology that involves identifying objects in images for various applications such as surveillance, security, safety and military operations. Object detection in low-illumination environments is a challenging task due to the low quality of the images. With the advancements in deep learning algorithms, object detection has become more precise and accurate. However, for nighttime, the traditional methods for object detection using visible light cameras are limited as visible light cameras often struggle to capture images in low light illumination, leading to limited visibility and increasing difficulties in object detection. On the other hand, thermal imaging can detect infrared radiation emitted by objects, providing clear images in the dark. Thermal imaging can be used for night vision detection because it captures the heat signature of objects instead of visible light, making it an effective solution for low light illuminations. Unlike visible light cameras, thermal imaging cameras are not affected by darkness, fog or other atmospheric conditions that can obstruct the view. Instead, they can provide a clear image of the environment and the objects within it, even in complete darkness.

In this research paper, we propose a method that can use deep learning algorithms for object detection in night vision using thermal infrared images. We suggest developing a model that can accurately detect objects in low light illuminations, making it useful for a wide range of applications [1]. We will implement deep learning algorithms such as YOLO (You Only Look Once), SSD (Single Shot Detector) and RetinaNet, particularly for night vision using thermal infrared images and compare them using various performance metrics, including accuracy, precision and recall. This proposal will provide insights into the effectiveness of using deep learning algorithms for object detection in night vision using thermal infrared images.

2. Related Work

Muhammad Javed Iqbal et al. [2] stated how feature-based FasterRCNNs like SqueezeNet, GoogleNet, ResNet-18, and ResNet-50 are used in real-time air surveillance to increase security. This states quadcopter-based surveillance using the captured images for anomaly detection. This initiative stated a wide range of advantages like better accuracy, no human error due to automation, cost efficiency and monitoring of disaster-stricken areas.

Y. Xiao et al. [1] proposed a night vision detector using a feature pyramid network and context fusion network. The problem faced in detection due to dim light or less exposure has been solved by using infrared monitoring. Various object detectors perform inappropriately due to less exposure; hence RFBNet is used. To obtain accurate results, image quality enhancement methods are followed.

In the paper[3], a bibliometric analysis for object detection methodologies which comes under the domain of Deep learning, which can do end-to-end object detection using Convolutional Neural Networks, is done. Algorithms designed for object detection are based on two approaches: one-stage object detection and two-stage object detection.

One-stage detectors have high inference speeds, and two-stage detectors have high localization and recognition accuracy. One of the reasons why object detection is the most valued field of research today is that visual information is widely spread across the world, making it important to advance the field of object detection for more accurate analysis.

Heena Patel and Kishor P. Upla [4] introduced an idea which is based on the utilization of thermal and visible spectrum pairs situated in an environment of the object for night vision surveillance. The authors proposed a network including fusion and MRCNN modules in which fusion modules use an encoder and decoder module with depthwise convolution to extract high-level features from input images and then, after this fused image is used to detect objects accurately. Experiments have been conducted on various datasets, and performance is verified for real-time night vision images. This shows that the proposed object detection method performs better than other state-of-the-art existing methods.

K. R. Akshatha et al.[5] evaluated the performance of Faster R-CNN and single-shot multi-box detector (SSD) algorithms for detecting humans in aerial thermal imagery. The study used two standard aerial thermal datasets and considered different backbone networks (ResNet50, Inception-v2, and MobileNet-v1). The results showed that Faster R-CNN with ResNet50 had the highest detection accuracy, with 100% mAP for the OSU thermal dataset and 55.7% for the AAU PD T dataset. The SSD with MobileNet-v1 had the highest detection speed of 44 FPS. Fine-tuning anchor parameters improved the mAP by 10% for Faster R-CNN ResNet50 and 3.5% for SSD Inception-v2. The study demonstrated the feasibility of using Faster R-CNN and SSD for human detection in aerial thermal imagery.

Mate Kristo et al. [6] investigated the use of convolutional neural network models for automatic person detection in thermal images. These models are becoming an important component in video surveillance systems due to their ability to perform well at night and in adverse weather conditions. The standard object detectors such as Faster R-CNN, SSD, Cascade R-CNN, and YOLOv3 were restrained on a dataset of thermal images and compared. YOLOv3 was found to be the fastest with comparable performance to the best and was used for further experiments. The minimum number of images needed for good detection results was determined, and the model was tested on different widely used thermal imaging datasets. The results of human and animal recognition in thermal images were also presented. The authors presented their original thermal dataset used for experimentation.

In 2021, Mohanad Al-Hasanat et al. [7] focused on using RetinaNet, an object detection deep learning model, to estimate the distance between objects in an image. The authors modified the RetinaNet architecture to include the estimation branch in addition to the object detection branch. This allowed the model to estimate the distance of objects in the image, along with detecting and classifying them. The results of their experiments showed that the model could be useful in various real-world applications such as autonomous vehicles and security due to its accuracy and the approach being relatively inexpensive and simple.

3. Methodology

The idea proposed in this paper focuses on building robust models using deep learning algorithms for object detection in night vision. To find an accurate algorithm that can detect objects using thermal infrared images, we analysed different object detectors based on their performance parameters. Faster R-CNN, MRCNN[4], HOG[8] , YOLO V3[9] , YOLO V8, SSD[10], RetinaNet[7] achieved best results for object detection. We selected one-stage detectors, namely YOLO, SSD and RetinaNet, as they have high inference speeds[3].

3.1. Data Collection and Pre-processing

We will gather different TIR(Thermal Infrared) videos, and further, they will be converted into image frames. Then these image frames can be resized according to different object detectors. YOLO keeps the aspect ratio safe without the need for explicit image resizing. Further data augmentation can be done on the resized images. Data augmentation is a technique where the size of the training dataset is increased by creating modified versions of existing data samples. The purpose of data augmentation is to reduce overfitting and to increase the robustness and generalisation of the trained model by exposing it to a wider range of variations in the input data. Common methods for data augmentation include flipping, rotation, scaling, cropping and adding noise to images.

3.2. YOLOv8

YOLO (You Only Look Once) is a popular object Detection algorithm that uses a single convolutional neural network to perform object detection and classification. Many versions of YOLO have been proposed, and in this paper, we have considered YOLOv8, the latest version of YOLO. YOLOv8 was developed by Ultralytics on 10 January 2023. It performs three tasks, namely object detection, image classification and instance segmentation. According to Ultralytics, YOLOv8 is fast, accurate, and easy to use, and it can be trained on large datasets. It has a new backbone network, a new anchor-free detection head, and a new loss function.

The YOLOv8 detector can be tested in isolated areas using thermal infrared images in this experiment. As mentioned above, data augmentation is used for increasing the accuracy of the detector, but in YOLOv8, there is no

need for data augmentation as it augments images during training automatically. Then we will annotate the images in the dataset using the LabelImg tool. LabelImg is open-source software that allows users to annotate images by drawing bounding boxes around objects in the image and then labelling those objects. After annotating the images, the annotations can be saved in YOLO format. We will obtain .txt files for every image. Each txt file will contain the class of object, its height, weight, x coordinate and y coordinate. Once the images are annotated, create a .yaml file which consists of the paths of the training and validation folder, the number of classes in the dataset and their names.

Split the dataset along with its corresponding .txt files into 80%, 10%, and 10% for training, validation and testing, respectively. For implementing YOLOv8, the prerequisite is installing and importing Ultralytics, as YOLOv8 can be imported from it. Once all necessary things are done, the model can be trained. For training, it is essential to specify the hyperparameters, such as the number of epochs, image size, task, etc. After the model is trained, we can also obtain the time required for training per image which can be used as a parameter for comparing the object detectors. The validation can be performed on the 10% of the dataset, and confusion matrix, f1 score, recall curve and other evaluation parameters can also be obtained. These evaluation results can be saved for further analysis. Later, the model can be tested by providing some thermal infrared images, and the output can be saved. We can also provide a video for testing. The output will be an image or video with bounding boxes to the objects along with its class probability.

3.3. Single Shot Detector(SSD)

A Single Shot Detector is an object detector based on VGG16 architecture. Single Shot Detector uses bounding boxes of various aspect ratios, compares them with ground truth boxes to obtain confidence scores and extracts feature maps.

SSD uses VGG16 for the extraction of the feature map. The input image is modified by adding the ground truth boxes around all the objects present in the image. These images are fed into the VGG16. In VGG16, there are thirteen convolutional layers, five Max Pooling layers, and three Dense layers though it has only sixteen weight layers. The convolution layers are of a 3x3 filter with stride 1 and always use the same padding and maxpool layer of a 2x2 filter. Max pooling reduces the feature map dimension. VGG16 performs strongly and does image classification tasks.

The architecture of the SSD consists of 6 convolutional layers preceded by VGG16, as shown in Fig.1. This set of auxiliary convolution layers helps feature extraction at multiple scales. It ultimately decreases the input size to the next corresponding layers[10]. For each object, we get 8732 bounding boxes. From these 8732 bounding boxes top 200 predictions are made based on the calculated confidence score of each box. The goal is to find the box that best fits the ground truth box. Intersection over Union (IoU) is calculated for the predicted bounding boxes and the ground truth boxes.[27]. Non-max suppression is used to filter these boxes and remove the duplicates. [12]

SSD takes input images with the ground truth boxes. SSD achieves better accuracy using various aspect ratios than other object detectors like YOLO. SSD has various applications like video forensics, legal investigations, landmark detections, and many more. In order to start implementation, we will use the LabelImg tool for annotation to obtain the images with ground truth boxes and the corresponding .xml files for each image. The collected thermal infrared images will be split into train, validation and test set into ratios of 80%, 10% and 10%, respectively—Import Tensorflow object detection API. Create a .csv file storing the details of the .xml files in comma-separated format for each training and validation set. Label_map.pbtxt file will be automatically generated, which stores the classes using the .xml files. Upload the generate_tfrecord.py file and generate TFrecords for each train and validation set. TFRecord format is a simple format a sequence of binary records. Load the tensorboard and train the model. Then validation is done using 10% data and finds confusion matrix, f1 score, recall curve and other evaluation parameters. Save the results of the evaluation. After validation, test the trained object detection model based on the test images, record the outputs and save them along with the bounding box and its class probability.

3.4. RetinaNet

RetinaNet is a single-shot object detection model developed by Facebook AI Research (FAIR). A deep neural network architecture extracts features from an image using a backbone network. Then it uses two separate branches to predict the class probabilities and the bounding box locations for each object in the image. RetinaNet made improvements over existing single-stage object detection models, i.e. Feature Pyramid Networks(FPN)[13] and Focal Loss. The various object detectors evaluate only a few locations of an image, and background objects are left, which leads to class imbalance problems. RetinaNet object detector uses Focal Loss to fill in for the class imbalances and inconsistencies, thus improving the speed of RetinaNet.

As shown in Fig.2, the architecture of RetinaNet breaks down into three components[14]:

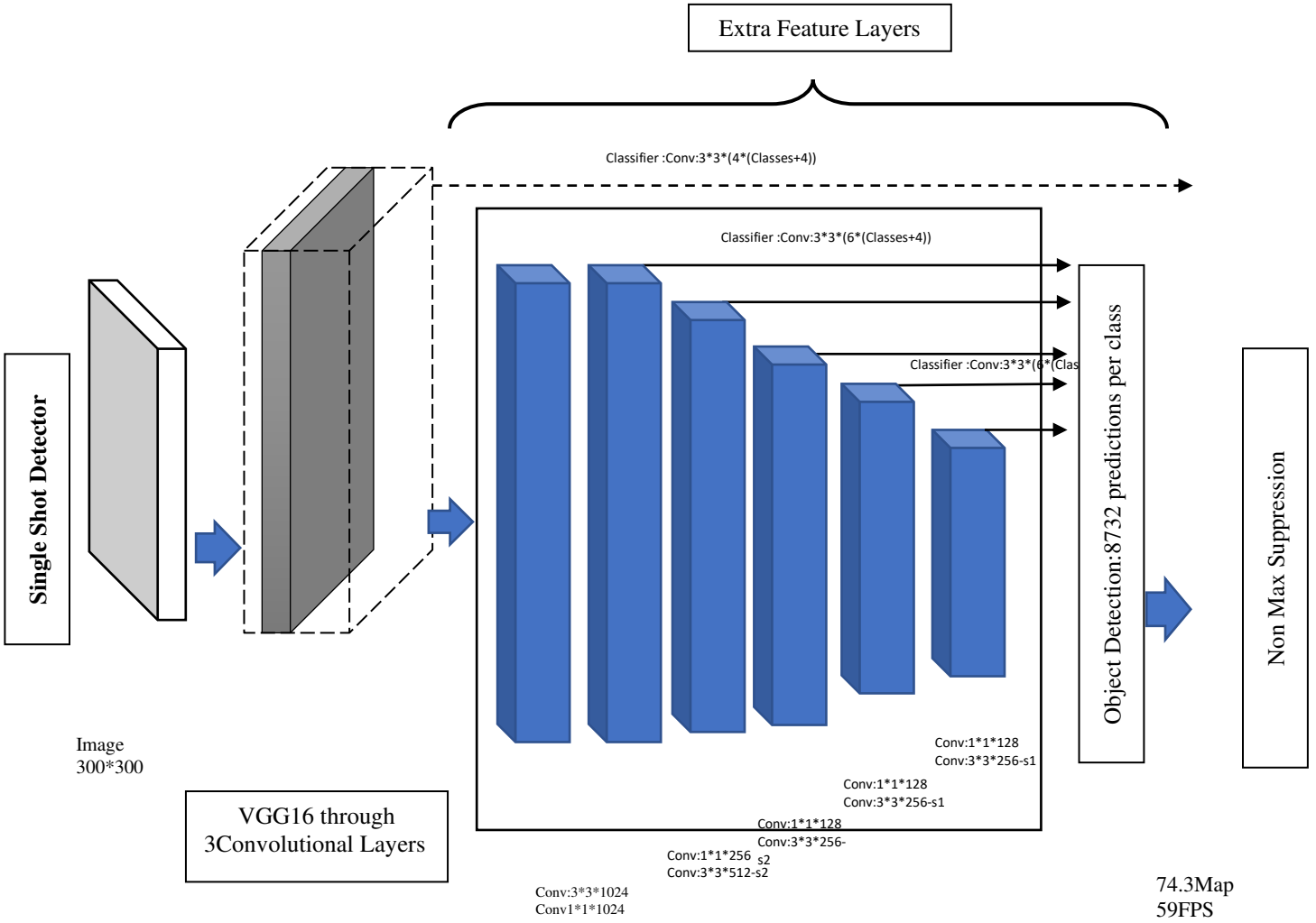

Fig. 1 Architecture of Single Shot Detector

Backbone Network(Bottom-up pathway + Top down the pathway with lateral connections)

Bottom-up pathway used in RetinaNet is Resnet. It is used for feature extraction, where the network starts from lower-level features and builds up to more complex features through multiple layers. This allows the network to learn the hierarchy of features.

3.4.1. Top-down Pathway with Lateral Connections

The top-down pathway means that the network starts with high-level features and refines them through multiple layers to obtain the final object detection. The lateral connections allow information to flow between different layers of the network, enabling the network to make predictions based on both high-level and low-level features. This top-down and lateral connections combination helps RetinaNet balance the trade-off between accuracy and computational efficiency. For this, RetinaNet uses Feature Pyramid Network(FPN).

3.4.2. Sub-network for Object Classification

In RetinaNet, the sub-network responsible for object classification is called the "classification subnet". The classification subnet takes the feature maps generated by the backbone network. It produces a set of class scores for each anchor box, indicating the likelihood of each anchor box containing a specific object class.

The classification subnet typically consists of several fully connected (FC) layers, sometimes preceded by a few convolutional (Conv) layers to reduce the number of feature channels and increase the spatial resolution.

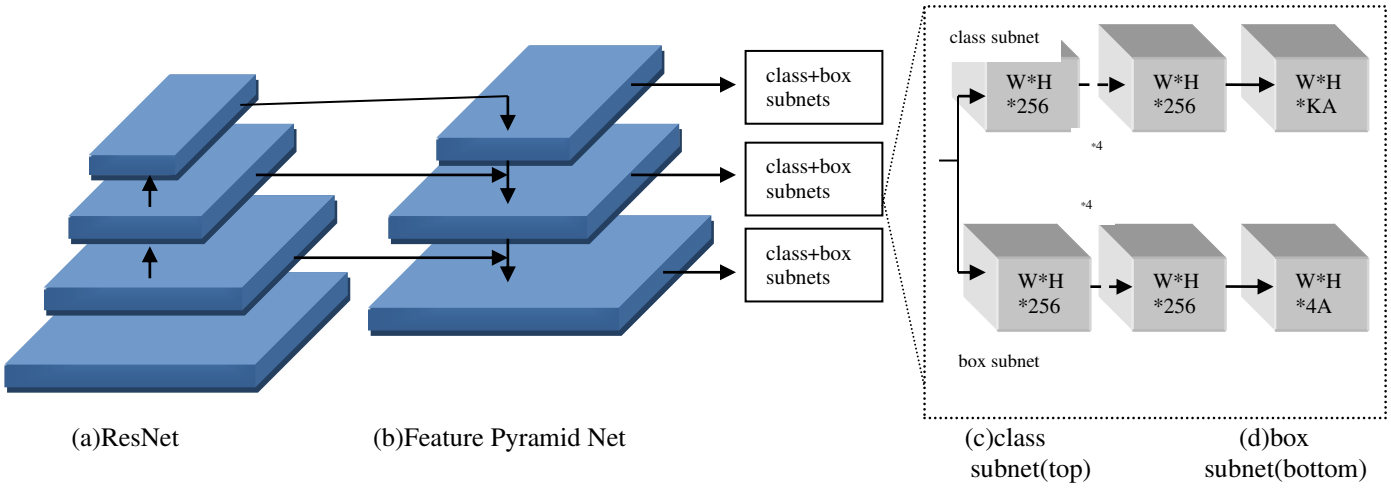

Fig. 2 Architecture of RetinaNet

The final FC layer produces the class scores, which are used along with the anchor boxes and the bounding box regression subnet to produce the final object detection results.

3.4.3. Sub-network for Object Regression

In RetinaNet, the regression sub-network is parallel Network (FPN) in a parallel manner. The sub-network for object regression is a fully connected layer that inputs features extracted by the Feature Pyramid Network (FPN). It outputs the bounding box coordinates for each object in the image. The sub-network uses a combination of anchor boxes and regression to predict the object-bounding boxes. The anchor boxes serve as prior knowledge, and the regression values refine the anchor boxes to be more accurate. The loss function used to train the sub-network penalises predicted bounding boxes far from the ground-truth boxes, encouraging the network to learn to generate accurate predictions.

For implementing RetinaNet, the first step that will be carried out is annotating the dataset using the LabelImg tool. Along with the YOLO format, the annotations can also be saved as .xml files in PASCAL VOC format, which RetinaNet needs. Once the images are annotated, we will create a pascal_voc.py file and define the dataset classes in the file. Later, we will split the dataset into training, validation and testing in an 80:10:10 ratio. Then we will produce train.txt and test.txt files containing lists of train and test files, respectively. After this step, we will download the pretrained weights, and the model can be trained. The argument we can pass while training the model can be batch-size, steps, epochs, weights, etc. Once the model is trained, we can get a weight file. Load it and build an inference model. Afterwards, the model will detect and evaluate objects in the images.

4. Conclusion

YOLOv8, SSD, and RetinaNet being one-stage detectors, may increase the speed of object detection. Furthermore, the models in this paper can be implemented, and the performance of these models can be compared to get the best object detector model for night vision. Also, the best model can be used for real-time object detection. Presently, we are working on the implementation of these models, and the results will soon be published in our next paper.

References

[1] Yuxuan Xiao et al., "Making of Night Vision: Object Detection Under Low-Illumination," *IEEE Access*, vol. 8, pp. 123075-123086, 2020. *Crossref,* https://doi.org/10.1109/ACCESS.2020.3007610

[2] Muhammad Javed Iqbal et al., "Real-Time Surveillance Using Deep Learning," *Security and Communications Networks*, 2021. *Crossref,* https://doi.org/10.1155/2021/6184756

[3] Aditya Lohia et al., *Bibliometric Analysis of One-stage and Two-stage Object Detection*, University of Nebraska-Lincoln, 2021.

[4] Heena Patel, and Kishor P. Upla, "Night Vision Surveillance: Object Detection using Thermal and Visible Images," *2020 International Conference for Emerging Technology (INCET)*, 2020. *Crossref,* https://doi.org/10.1109/INCET49848.2020.9154066

[5] K. R. Akshatha et al., "Human Detection in Aerial Thermal Images Using Faster R-CNN and SSD Algorithms," *Electronics*, vol. 11, no. 7, p. 1151, 2022. *Crossref,* https://doi.org/10.3390/electronics11071151

[6] Mate Kristo, Marina Ivasic-Kos, and Miran Pobar, "Thermal Object Detection in Difficult Weather Conditions Using YOLO," *IEEE Access*, vol. 8, pp. 125459-125476, 2020. *Crossref,* https://doi.org/10.1109/ACCESS.2020.3007481

[7] Mohanad Al-Hasanat et al., "RetinaNet-based Approach for Object Detection and Distance Estimation in an Image," *International Journal on Communications Antenna and Propagation (IRECAP)*, vol. 11, no. 1, p. 19, 2021. *Crossref,* http://dx.doi.org/10.15866/irecap.v11i1.19341

[8] Navneet Dalal, and Bill Triggs, "Histograms of Oriented Gradients for Human Detection," *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2005. *Crossref,* https://doi.org/10.1109/CVPR.2005.177

[9] Qasem Abu Al-Haija, Manaf Gharaibeh, and Ammar Odeh, "Detection in Adverse Weather Conditions for Autonomous Vehicles via Deep Learning," *AI Journal*, vol. 3, no. 2, pp. 303-317, 2022. *Crossref,* https://doi.org/10.3390/ai3020019

[10] Wei Liu et al., "SSD: Single Shot MultiBox Detector," *Computer Vision and Pattern Recognition*, 2015. *Crossref,* https://doi.org/10.48550/arXiv.1512.02325

[11] M. Maheswari, M. S. Josephine, and V. Jeyabalaraja, "YOLO Architecture-based Object Detection for Optimizing Performance in Video Streams," *International Journal of Engineering Trends and Technology*, vol. 70, no. 11, pp. 187-196, 2022. *Crossref*, https://doi.org/10.14445/22315381/IJETT-V70I11P220

[12] Jan Hosang, Rodrigo Benenson, and Bernt Schiele, "Learning Non-maximum Suppression," *Computer Vision and Pattern Recognition,* 2017. *Crossref,* https://doi.org/10.48550/arXiv.1705.02950

[13] Tsung-Yi Lin et al., "Feature Pyramid Networks for Object Detection," *Computer Vision and Pattern Recognition,* 2016. *Crossref,* https://doi.org/10.48550/arXiv.1612.03144

[14] Tsung-Yi Lin et al., "Focal Loss for Dense Object Detection," *Computer Vision and Pattern Recognition,* 2017. *Crossref,* https://doi.org/10.48550/arXiv.1708.02002

[15] Aashish Bhandari et al., "Image Enhancement and Object Recognition for Night Vision Surveillance," *ICTRCET 18 at Bangaluru*, 2018.

[16] Rupesh P.Raghatate et al, "Night Vision Techniques and Their Applications," *International Journal of Modern Engineering Research (IJMER)*, vol. 3, no. 2, pp. 816-820, 2013.

[17] D. Malarvizhi, V. Lavanya, and M. Nivetha priya, "Night Vision Technology," *International Journal for Scientific Research and Development*, vol. 3, no. 8, 2017.

[18] Kai Hu et al., "A Marine Object Detection Algorithm Based on SSD and Feature Enhancement," *Complexity*, 2020. *Crossref,* https://doi.org/10.1155/2020/5476142

[19] Sreehari Patibandla, Maruthavanan Archana, and Rama Chaithanya Tanguturi, "Object Tracking using Multi Adaptive Feature Extraction Technique," *International Journal of Engineering Trends and Technology*, vol. 70, no. 6, pp. 279-286, 2022. *Crossref*, https://doi.org/10.14445/22315381/IJETT-V70I6P229

[20] Pavan Sai Vemulapalli et al., "Multi-object Detection in Night Time," *Asian Journal of Convergence in Technology*, vol. 5, no. 3, pp. 1-7, 2019.

[21] Tanvir Ahmad et al., "Object Detection through Modified YOLO Neural Network," *Scientific Programming,* 2020. *Crossref,* https://doi.org/10.1155/2020/8403262

[22] Muskan Choudhary et al., "Object Detection Using YOLO Models," *International Research Journal of Engineering and Technology,* vol. 9, no. 5, pp. 3785-3789, 2022.

[23] Joaquın Royo Miquel et al., "RetinaNet Object Detector based on Analog-to-Spiking Neural Network Conversion," *Image and Video Processing,* 2021. *Crossref,* https://doi.org/10.48550/arXiv.2106.05624

[24] R. Manasa, K Karibasappa, and J. Rajeshwari, "Autonomous Path Finder and Object Detection using an Intelligent Edge Detection Approach," *SSRG International Journal of Electrical and Electronics Engineering*, vol. 9, no. 8, pp. 1-7, 2022. *Crossref*, https://doi.org/10.14445/23488379/IJEEE-V9I8P101

[25] Tsung-Yi Lin et al., "Feature Pyramid Networks for Object Detection," *in Processing IEEE Conference on Computer Visison Pattern Recognition (CVPR)*, 2016. *Crossref,* https://doi.org/10.48550/arXiv.1612.03144

[26] Tsung-Yi Lin et al., "Feature Pyramid Networks for Object Detection," *in Processing IEEE Conference on Computer Visison Pattern Recognition (CVPR),* 2016. *Crossref,* https://doi.org/10.48550/arXiv.1612.03144

[27] Wu Zheng et al., "CIA-SSD: Confident IoU-Aware Single-Stage Object Detector from Point Cloud," *in Processing IEEE Conference on Computer Visison Pattern Recognition (CVPR),* 2020. *Crossref,* https://doi.org/10.48550/arXiv.2012.03015

[28] Amit Tiwari, and Jalaj Gupta, "A Simulation of Night Vision Technology Aided with AI," *AKGEC International Journal of Technology*, vol. 12, no. 1, 2022.

[29] Pranav Adarsh, Pratibha Rathi, and Manoj Kumar, "YOLO v3-Tiny: Object Detection and Recognition using One Stage Improved Model," *2020 6th International Conference on Advanced Computing and Communication Systems*, 2020. *Crossref,* https://doi.org/10.1109/ICACCS48705.2020.9074315