[ICAETBM-2024]
Print ISSN: 2395-1990 | Online ISSN : 2394-4099 (www.ijsrset.com)
doi : https://doi.org/10.32628/IJSRSET2411592

Night time Surveillance in Communication Using YOLOv3

Tapas K. Madan Pramanik[1], Prakash Gajananrao Burade[1], Sanjeev Sharma[2*]

[1]Department of Electrical & Electronics Engineering, Faculty of Engineering, Sandip University, Nashik, Maharashtra, India

[2*]Department of Electronic and Communication Engineering, New Horizon College of Engineering, Bangalore, Karnataka, India

ABSTRACT

Nighttime surveillance presents significant challenges due to low visibility, varying lighting conditions, and interference from artificial light sources. Hence, this study proposed an effective object detection framework using the YOLOv3 model to enhance real-time monitoring in night surveillance applications, specifically within communication and security systems. YOLOv3's architecture, with its multi-scale detection and use of predefined anchor boxes, enables robust detection of objects under low-light environments and amidst light interference from vehicles and streetlights. The proposed system is tested on a night surveillance dataset, where it demonstrates high precision and speed in identifying objects, making it suitable for real-time applications. With Mean Average Precision (mAP) 87.9%, YOLOv3 effectively balances detection accuracy with inference time, ensuring minimal latency in live surveillance feeds. The results indicate that YOLOv3 outperforms traditional models such as Faster RCNN, particularly in detecting small objects under poor illumination. This approach offers a reliable solution for enhancing communication and security systems in nighttime surveillance scenarios.

Keywords : Mean Average Precision , YOLOv3, DL Models, Deep Learning, Single Shot Detector

I. INTRODUCTION

With the continuous advancement of computational power, Deep Learning (DL) models have become increasingly prominent in the field of object detection [1-3]. These methods have gradually emerged as the mainstream approach for detecting objects in images. DL models mimic the human brain's visual perception system by extracting features directly from raw images and processing them through multiple layers to capture high-dimensional information[4-7].

Currently, there are two primary categories of DL models for object detection. The first category involves a two-stage process, separating object detection into a candidate box selection stage and an object classification stage like R-CNN series and its derivatives [8-11]. The second category consists of single stage models that treat classification and bounding box regression as a unified task. Examples of these include Single Shot Detector (SSD) and You Only Look Once (YOLO).

However, both types of algorithms face challenges when it comes to detecting small objects. Under the standard definition, a small object is one that occupies 0.12% or less of a 256x256 pixel image. Detecting such objects can be difficult because the features extracted by the network from small objects are significantly fewer compared to larger objects, leading to suboptimal model performance [12-14].

Hence, this work focuses on improving the detection of small objects by enhancing the YOLOv3 network. First, feature enhancement is applied in the feature extraction module, using Darknet 53 in the backbone network. Additionally, a combination of loss function is utilization which is known for its strong generalization ability. These improvements collectively aim to increase the accuracy of small object detection.

II. REVIEW OF LITERATURE

Deep learning has its roots in traditional Artificial Neural Networks (ANNs) [7]. Object detection is a fundamental application of deep learning with widespread utility in areas such as autonomous driving and safety systems. It has achieved significant success in many fields, largely due to the availability of large datasets and the effectiveness of Convolutional Neural Networks (CNNs). Object detection algorithms can generally be divided into two categories:

1. **Two-stage algorithms**, which first generate candidate bounding boxes based on the input image and then classify the objects in these boxes using CNNs. Examples include Fast R-CNN and Faster R-CNN.
2. **Single-stage algorithms**, which treat detection as a regression problem and eliminate the need for generating candidate boxes. Examples of these algorithms include YOLO [4] and SSD [3].

YOLO, a single-stage detection algorithm, processes the entire image through the network, using CNNs to extract features from the whole image. It then performs regression to detect objects in a single step. Although Faster R-CNN [10] reduces the computational cost associated with sliding windows, it is still constrained by the use of fixed-size windows. In contrast, YOLO divides the image into non-overlapping grid cells, avoiding the need for numerous sliding windows and thus significantly increasing detection speed.

The YOLO network undergoes preliminary training on ImageNet before the main training phase. This is followed by four randomly initialized convolutional layers and two fully connected layers. The pre-trained model consists of 20 convolutional layers, an average pooling layer, and a fully connected layer. YOLO predicts an $S \times S \times B$ bounding box grid, which is far fewer than the thousands of sliding windows used in two-stage detection algorithms. This greatly improves detection speed, although it results in a slight decrease in detection accuracy [11].

III. METHODOLOGY

The methodology adopted in this work is highlighted in Figure 1

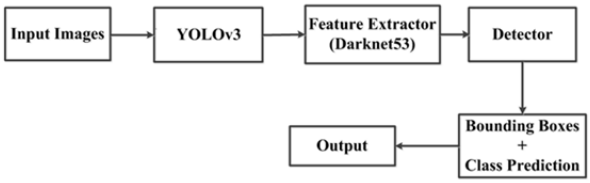


Figure 1. Block diagram - Proposed Topology

3.1. YoloV3

YOLOv3 treats object detection as a regression task, where it directly predicts class probabilities and bounding box offsets from the entire image in a single forward pass through a convolutional neural network. Unlike traditional methods, it entirely eliminates the need for region proposal generation and feature resampling, consolidating all detection stages within a single network. This design enables YOLOv3 to function as a true end-to-end detection system. Thus, the architecture of the yolo v3 is depicted in figure 2.

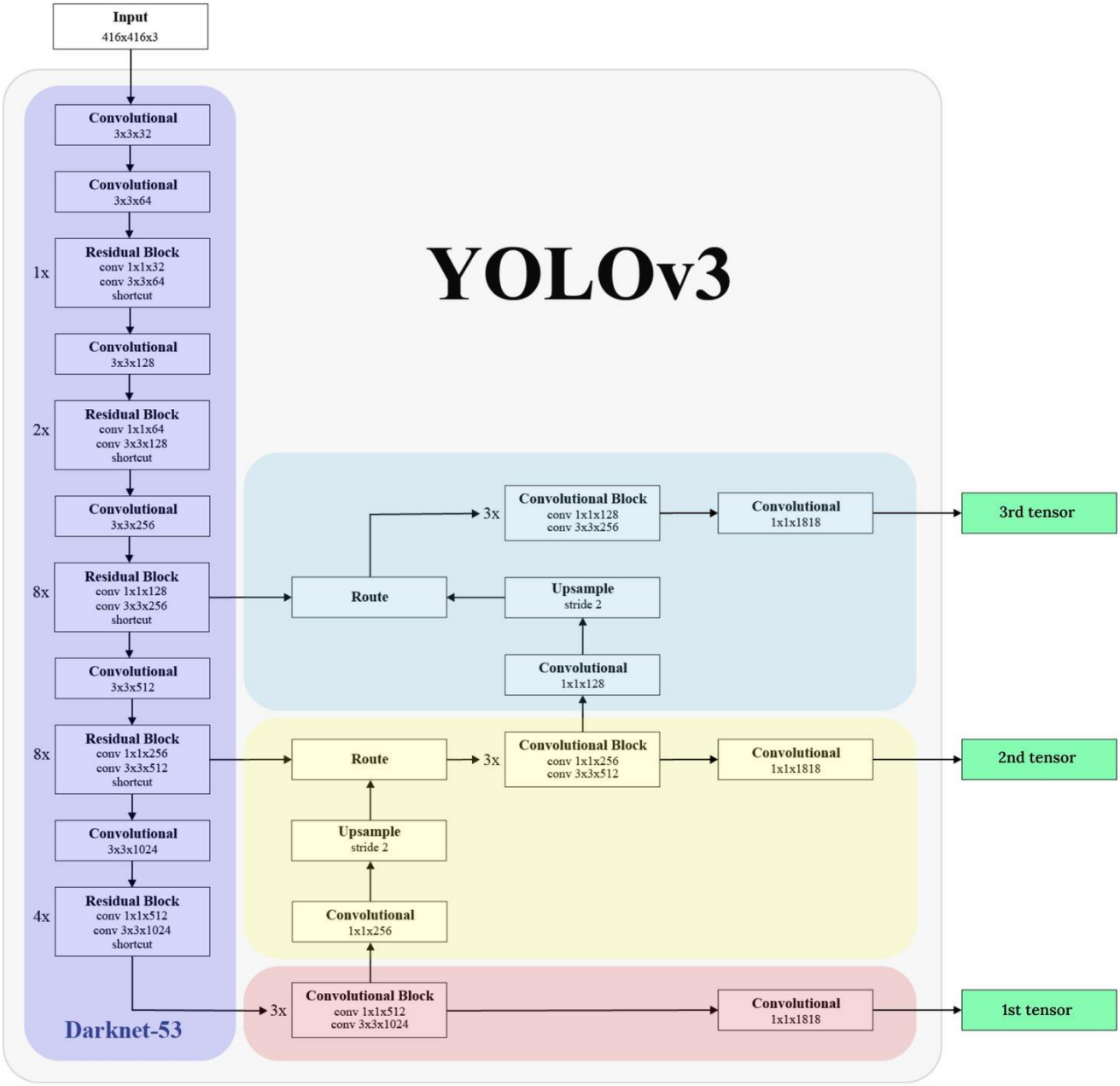


Figure 2. Architecture of the yolo v3

Thus, an overview of the architecture of YOLOv3 is as follows

3.2. Backbone: Darknet-53

YOLOv3 uses a feature extractor called **Darknet-53** as its backbone network. It is deeper and more powerful than the Darknet-19 used in YOLOv2. Darknet-53 features 53 convolutional layers, making it deeper and more powerful. This increased depth enhances the network's ability to capture complex features, boosting its detection performance.

The architecture shown in figure 3 adopts a modular design, where each module contains a series of convolutional layers coupled with shortcut connections.

	Type	Filters	Size	Output
	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3 / 2	128 × 128
1 ×	Convolutional	32	1 × 1	
	Convolutional	64	3 × 3	
	Residual			128 × 128
	Convolutional	128	3 × 3 / 2	64 × 64
2 ×	Convolutional	64	1 × 1	
	Convolutional	128	3 × 3	
	Residual			64 × 64
	Convolutional	256	3 × 3 / 2	32 × 32
8 ×	Convolutional	128	1 × 1	
	Convolutional	256	3 × 3	
	Residual			32 × 32
	Convolutional	512	3 × 3 / 2	16 × 16
8 ×	Convolutional	256	1 × 1	
	Convolutional	512	3 × 3	
	Residual			16 × 16
	Convolutional	1024	3 × 3 / 2	8 × 8
4 ×	Convolutional	512	1 × 1	
	Convolutional	1024	3 × 3	
	Residual			8 × 8
	Avgpool		Global	
	Connected		1000	
	Softmax			

Figure 3. Architecture – Darkness 53

Features of Darknet-53:

- **53 Convolutional Layers**: The backbone is composed of 53 convolutional layers. It uses only 3x3 and 1x1 filters, which are efficient and suitable for feature extraction.
- **Residual Connections**: Similar to ResNet, Darknet-53 utilizes residual connections to ease the training of deeper networks and improve gradient flow. This allows the model to avoid vanishing gradient issues while keeping computational complexity relatively low.
- **No Fully Connected Layers**: Like its predecessors, YOLOv3 does not use fully connected layers, making it more computationally efficient.
- **Conv-BatchNorm-Leaky ReLU blocks**: Repeated sequences of convolutional layers followed by batch normalization and leaky ReLU activation.
- **Downsampling**: Achieved through convolutional layers with a stride of 2, reducing the spatial dimensions while increasing depth.

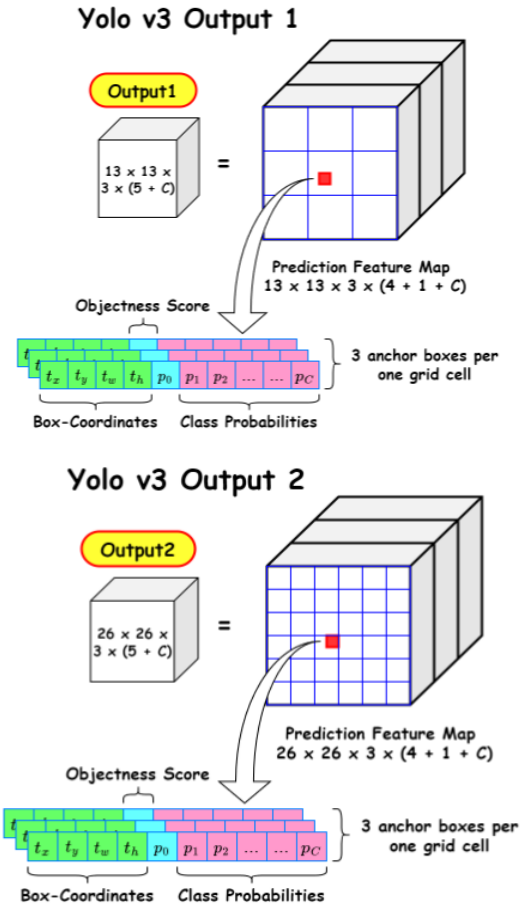
3.3. Detection Head

YOLOv3 performs detection at three different scales, allowing it to better detect both large and small objects. This is one of the major improvements over YOLOv2, where only a single scale was used.

Multiscale Detection:

- YOLOv3 extracts features at three different scales from the Darknet-53 backbone, corresponding to three different levels of abstraction.
- These feature maps are taken from different depths within the network:
  - **First scale**: A detection is made on a 13x13 grid (downsampled by a factor of 32).
  - **Second scale**: A detection is made on a 26x26 grid (downsampled by a factor of 16).
  - **Third scale**: A detection is made on a 52x52 grid (downsampled by a factor of 8).

Each scale can detect objects at different sizes, with the 13x13 grid better suited for large objects, the 26x26 grid for medium objects, and the 52x52 grid for small objects and is depicted in figure 4.



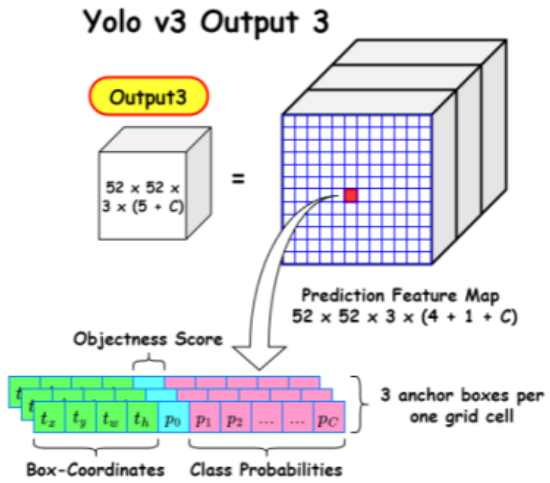


Figure 4. Feature maps

3.4. Bounding Box Prediction

For each grid cell in the three detection heads, YOLOv3 predicts:

- **Bounding boxes**: YOLOv3 uses **anchor boxes** for bounding box prediction, where it predicts 4 coordinates: x,y,w,hx, y, w, hx,y,w,h (center coordinates, width, and height of the bounding box).
- **Objectness score**: This score tells how likely it is that an object exists within the bounding box. It is essentially a binary classification (object/no-object).
- **Class probabilities**: For each bounding box, YOLOv3 predicts the probability distribution over all possible classes. YOLOv3 can handle **multiple classes** in the same image.

Each grid cell predicts 3 bounding boxes using **predefined anchor boxes**, so at each scale, the network predicts a total of 3 bounding boxes.

3.5. Loss Function

YOLOv3 uses a combination of loss functions for object detection:

- **Localization Loss**: This measures the accuracy of the predicted bounding box coordinates compared to the ground truth.
- **Confidence Loss (Objectness Loss)**: This penalizes the network if it predicts an object where there is none, or if it fails to predict an object that is present.
- **Class Prediction Loss**: This penalizes incorrect predictions of object classes. Thus, the loss function of Yolov3 can be depicted as follows

$$\lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right]$$

$$+ \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right]$$

$$+ \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left(C_i - \hat{C}_i \right)^2$$

$$+ \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{noobj}} \left(C_i - \hat{C}_i \right)^2$$

$$+ \sum_{i=0}^{S^2} \mathbb{1}_{i}^{\text{obj}} \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2$$

3.6. Feature Maps

YOLOv3 uses predefined **anchor boxes** for bounding box prediction. These anchor boxes help the model to make better predictions for objects of varying shapes and sizes. At each detection scale, YOLOv3 predicts 3 bounding boxes per grid cell, leading to a total of 9 anchor boxes across the 3 scales.

The predefined anchor box dimensions are clustered on the COCO dataset, and they are:

- **Small**: (10x13, 16x30, 33x23) for fine-scale feature maps (52x52 grid).
- **Medium**: (30x61, 62x45, 59x119) for mid-scale feature maps (26x26 grid).
- **Large**: (116x90, 156x198, 373x326) for coarse-scale feature maps (13x13 grid).

3.7. Activation Function: Sigmoid

- **Sigmoid**: YOLOv3 applies the sigmoid function to the bounding box predictions and objectness scores, so they are bounded between 0 and 1.

This ensures that the bounding box predictions (center coordinates and width/height) are constrained within the bounds of the grid cell, and the objectness score is a probability between 0 and 1.

3.8. Class Predictions

It uses independent logistic classifiers for each class. This means each class prediction is treated independently, which makes it better at handling overlapping classes or multilabel classification.

IV. RESULTS AND DISCUSSION

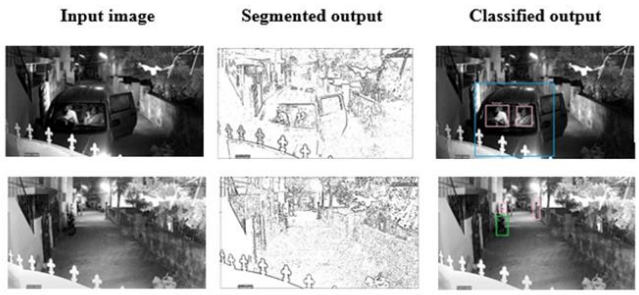


Figure 5. Segmented and classified output using YOLO V3

The trained classifier can be applied to new data, where samples from various categories are classified into their respective groups. Several parameters were analyzed to assess the effectiveness of the approach, with the performance evaluated using a range of metrics.

One of the key metrics used is **Mean Average Precision (mAP),** which serves as a standardized measure of the model's performance in object detection tasks, often referred to simply as AP. Finally, the results were compared with alternative algorithms and detection methods, based on key performance metrics.

Table 1. Detection performance is expressed in %, and the detection speed is with ms

Methods	Time	mAP
Faster RCNN	190.2	65.02
YOLOv3	39.57	87.9

This section evaluates the effectiveness of the proposed approach. In this, the effectiveness of the proposed system is analysed using SSAN dataset. The YOLOv3 classifier is utilized for both training and testing purposes, as shown in Figure 5.

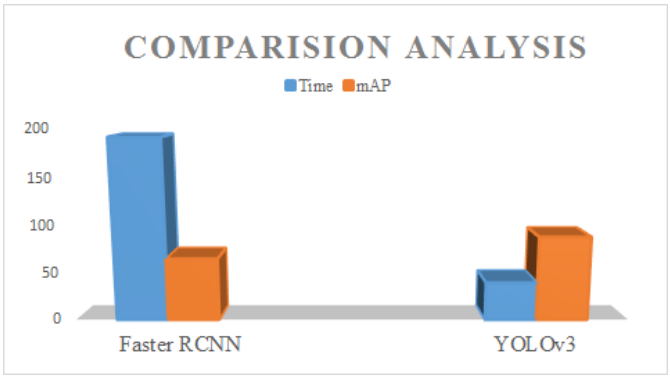


Figure 6. Comparison Analysis

The results indicate that, compared to contemporary models like Faster RCNN and YOLOv3, the proposed model (YOLOv3) outperforms them in terms of speed and accuracy in object detection, particularly in low-light conditions and when light interference from vehicles is present, as demonstrated in Figure 6 and summarized in Table 1.

Due to its superior performance, YOLOv3 was selected for nighttime object detection tasks. Despite challenges such as low resolution, significant noise, and limited information in infrared (IR) images from the SSAN dataset, YOLOv3 consistently achieved strong detection results under low illumination and light interference from vehicles. The architecture also exhibited high precision in object localization without compromising inference speed, making it reliable and efficient for detecting objects in night surveillance scenarios.

V. SUMMARY

YOLOv3 was developed and tested in street night surveillance scenarios, aimed at detecting and tracking

objects in typical low-light conditions using the SSAN dataset. A comparison of leading detection models, including Faster RCNN showed that YOLOv3 significantly outperformed them in terms of both speed and accuracy. The model achieved a mean Average Precision (mAP) of 87.9% with a processing time of 39.57 ms. This superior performance, especially in low illumination and light interference from vehicles, makes YOLOv3 a promising foundation for further development in night object detection tasks.

VI. REFERENCES

[1]. Shlhamer E, Long J, Darrell T. (2016) "Fully convolutional networks for semantic segmentation." IEEE Transactions on Pattern Analysis &Machine Intelligence 79(10):1337-1342.

[2]. Girshick R, Donahue J, Darrell T, et al. (2014) "Rich feature hierarchies for accurate object detection and semantic segmentation." IEEE Conference on Computer Vision and Pattern Recognition, USA: IEEE 580-587.

[3]. Liu W, Auguelov D, Erhan D, et al. (2016) "SSD: Single Shot MultiBox Detector." European Conference on Computer Vision (ECCV). San Francisco, CA, USA: IEEE Conference 6517-6525.

[4]. Redmon J, Divvals S, Grishick R, et al. (2016) "You Only Look Once: unified, real time object detection." IEEE Conference on Computer Vision and Pattern Recognition, USA: IEEE 779-788.

[5]. Zhang X Y, Ding Q H, Luo H B, et al. (2017) "Infrared dim target detection algorithm based on improved LCM." Infrared and Laser Engineering 46(7): 0726002.

[6]. Redmon J, Farhadi A. (2018) "YOLOv3: An Incremental Improvement." IEEE Conference on Computer Vision and Pattern Recognition. USA: IEEE 2311-2314.

[7]. Erhan D, Szegedy C, Toshev A, et al. (2014) "Scalable object detection using deep neural networks." Proceedings of the IEEE conference on computer vision and pattern recognition.2147-2154.

[8]. MCCULLOCH WS, PITTS W. (1943) "A logical calculus of the ideas immanent in nervous activity." The bulletin of mathematical biophysics 5(4):115-133.

[9]. Zeiler M D, Krishnan D, Taylor G W, et al. (2010)"Deconvolutional networks." Computer Vision & Pattern Recognition.

[10]. M. K. Pargi, B. Setiawan and Y. Kazama. (2019) "Classification of different vehicles in traffic using RGB and Depth images: A Fast RCNN Approach." 2019 IEEE International Conference on Imaging Systems and Techniques (IST), Abu Dhabi, United Arab Emirates 1-6.

[11]. Y Li, K He, and J Sun. (2016) "R-fcn: Object detection via region based fully convolutional networks." In Advances in Neural Information Processing systems, 630-645.

[12]. Z. Wang, Z. Cheng, H. Huang and J. Zhao, (2019) "ShuDA-RFBNet for Real-time Multi-task Traffic Scene Perception." 2019 Chinese Automation Congress (CAC), Hangzhou, China 305-310.

[13]. Sandler M, Howard A, Zhu M, et al. (2018) "MobileNetV2: inverted residuals and linear bottlenecks." IEEE Conference on Computer Vision and Pattern Recognition. Salt Lake City: IEEE 4510-4520.

[14]. Ioffe S, Szegedy C. (2015) "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shife." International Conference on Machine Learning. USA: ICML.