 doi:10.20944/preprints202204.0279.v1

*Article*

# Exploiting Concepts of Instance Segmentation to Boost Detection in Challenging Environments

**Khurram Azeem Hashmi [1,2,3]*, Alain Pagani [3], Marcus Liwicki [4], Didier Stricker [1,3] and Muhammad Zeshan Afzal [1,2,3]**

[1] Department of Computer Science, Technical University of Kaiserslautern, 67663 Kaiserslautern, Germany; khurram_azeem.hashmi@dfki.de (K.A.H.); muhammad_zeshan.afzal@dfki.de (M.Z.A); didier.stricker@dfki.de (D.S.)
[2] Mindgarage, Technical University of Kaiserslautern, 67663 Kaiserslautern, Germany
[3] German Research Institute for Artificial Intelligence (DFKI), 67663 Kaiserslautern, Germany; alain.pagani@dfki.de
[4] Department of Computer Science, Luleå University of Technology, 971 87 Luleå, Sweden; marcus.liwicki@ltu.se
* Correspondence: khurram_azeem.hashmi@dfki.de

**Abstract:** In recent years, due to the advancement of machine learning, object detection has become a mainstream task in the Computer Vision domain. The first phase of object detection is to find the regions where objects can exist. With the improvement of deep learning, traditional approaches such as sliding windows and manual feature selection techniques have been replaced with deep learning techniques. However, object detection algorithms face a problem when performing in low light, challenging weather, and crowded scenes like any other task. Such an environment is termed a challenging environment. This paper exploits pixel-level information to improve detection under challenging situations. To this end, we exploit the recently proposed hybrid task cascade network. This network works collaboratively with detection and segmentation heads at different cascade levels. We evaluate the proposed methods on three complex datasets of ExDark, CURE-TSD, and RESIDE and achieve an mAP of 0.71, 0.52, and 0.43, respectively. Our experimental results assert the efficacy of the proposed approach.

**Keywords:** object detection; challenging environments; low-light; image enhancement; complex environments; deep neural networks; computer vision

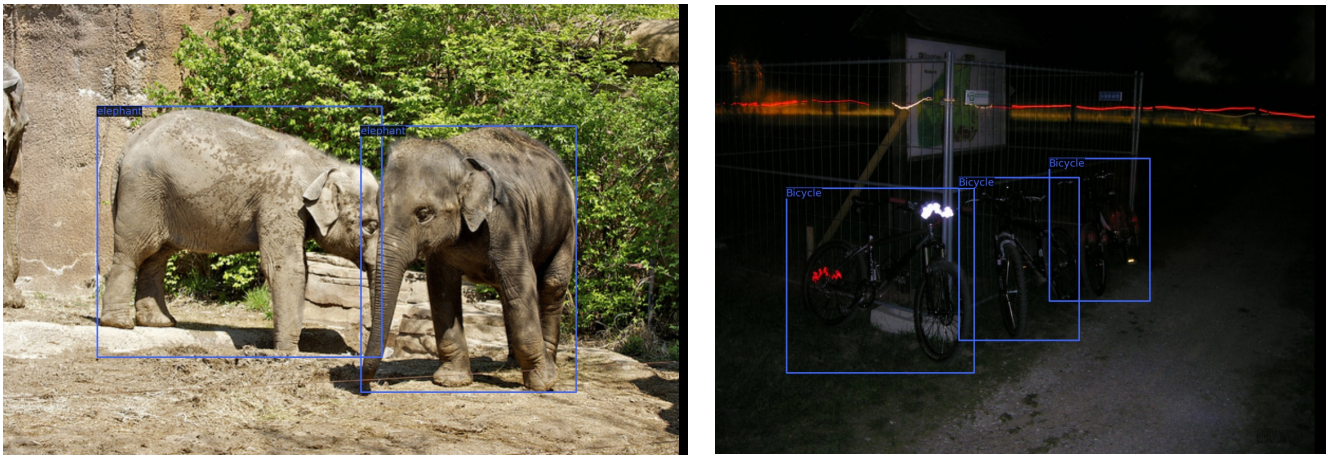
## 1. Introduction

One of the most important and widely used tasks in the field of computer vision is object detection. Over the years, many techniques have been employed to improve the performance of object detection. Object detection has various applications such as instance segmentation [1–3], visual question answering [4], image captioning [5,6], object tracking [7], activity recognition [8–10] and so on. The process of object detection can be broken down into the following steps: identifying the object and spatial localization of the object to provide exact coordinates of the object's location.

The environment of object detection algorithms is mainly categorized into two types [11], object detection in a general environment and object detection in a challenging environment. A general environment is rich in contextual features and has low object cluttering and occlusions. Compared to the general environment, a challenging environment is composed of low contextual features, object cluttering, various occlusions, and objects merged with the background. In real-time scenarios, it is frequent that the input images received by the object detection network are not spatially rich as they are captured in complex scenarios and have low-light conditions. In this paper, we have referred all these situations to a challenging environment.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 28 April 2022 doi:10.20944/preprints202204.0279.v1






**(a)** Generic Object Detection **(b)** Challenging Object Detection

**Figure 1.** Visual illustration of the different between object detection in a generic and challenging environment. Figure 1a is a sample image taken from the COCO dataset [12], whereas Figure 1b is taken from the ExDark datset [13]. The blue color represents ground truth annotation.

Recently, various approaches like a fusion of domains using glue layers [14], fusing thermal images with RGB images [15] and combination [16] of Deep Convolution Generative Adversarial Networks (DCGAN) [17] and Faster R-CNN [18] have been proposed to tackle the problem of object detection in challenging environments. These approaches improved the performance, but are dependent on image enhancement as a pre-processing step and prior assumptions about the type and shape of objects.

In one of the recent works, Ahmed et al. [11] investigated the capabilities of modern object detection algorithms on the datasets captured either in low illuminance environment or in harsh conditions. In this paper, by taking a step forward in this direction, we propose a framework that leverages pixel-level information by employing the powerful recently proposed Hybrid Task Cascade (HTC) network with a pre-trained ResNext-101 as a backbone network. The proposed pipeline is depicted in Figure 2.

To encapsulate, the main contributions of this work are explained below:

- This paper presents an end-to-end optimizable framework to tackle the problem of object detection under low illuminance and arduous conditions.
- We evaluate the proposed method on three different challenging datasets and achieve an mAP of 0.71, 0.52, and 0.43 on the datasets of ExDark, RESIDE, and CURE-TSD, respectively.
- Unlike previous works, the presented system does not rely on any pre-processing techniques such as image enhancement to accomplish the results.

The remaining article is organized as follows. Section 2 describes the prior literature dealing with both generic and challenging environments through traditional computer vision or statistical learning-based approaches. Section 3 talks about the presented object detection framework and describes the individual components.Section 4 presents the comprehensive overview of employed datasets. Section 5 explains the experimental details, evaluation metrics and presents a quantitative and qualitative analysis of the proposed system. Section 6 ends the paper with a brief conclusion and a discussion on the future work.

## 2. Related work

Previous work in the field of object detection can be distinguished into two categories, namely generic object detection and object detection in challenging environment [11]. Section 2.1 provides a brief overview of earlier approaches based on traditional computer vision algorithms to solve object detection in both generic and visually difficult

environment. Section 2.2 discusses learning-based mainly deep learning-based methods in both environments.

### *2.1. Traditional Approaches*

In the early days of computer vision [19], traditional algorithms used for object detection required handcrafted features and manual parameter tuning. Traditional algorithms can be categorized into approaches for the generic environment and the challenging environment.

#### 2.1.1. Generic Environment

The first traditional algorithm was Viola-Jones(VJ) detector [20], which used a sliding window approach to find objects. Later, more advanced algorithms like Histogram of Oriented Gradients (HOG) Detector [21] and Deformable Part-based Model [22] were introduced. Over the years, various surveys have been conducted on object detection in general environment [23–26] comparing different architectures from traditional to deep learning-based approaches, along with various datasets used as benchmarks to evaluate the performance of each algorithm [27].

#### 2.1.2. Challenging Environment

For challenging environments, traditional approaches for object detection employed template matching [28,29]. These approaches are difficult to extend to multiple classes, as for each object, a template is required. Later, Constantine et al. [30] proposed a method that uses wavelet representation with a support vector machine to detect objects in a given input image. The wavelet representation was calculated from statistical analysis of class instances. Another approach by Shirai et al. [31] for detecting objects required manual parameter tuning to find all objects and needed a few assumptions such as type and shape of an object prior to detection.

### *2.2. Machine Learning-Based Approaches*

Nowadays, deep learning-based algorithms are preferred as they automatically learn features and tune hyper parameters to find optimal results [32]. Like traditional approaches, learning-based approaches can be divided into two groups, learning-based approaches for generic environments and for challenging environments.

#### 2.2.1. Generic Environment

R-CNN [33] was the first learning-based network introduced in 2014 to solve the object detection problem. The network first extracted region proposals from the input image using selective search [34] and then combined them with Convolutional Neural Networks (CNN) to find objects. In 2015 Fast R-CNN [35] an improved version of R-CNN was proposed. Fast R-CNN passed the input image through CNN first to generate feature maps compared to its predecessor. Proposal regions were then selected from these generated feature maps using selective search. To take full advantage of resources, GoogleLeNet [36] was introduced after Fast R-CNN. Compared to the previous networks, GoogleNet architecture allowed an increase in the width and depth of the network while keeping computation low. Compared to traditional algorithms, these networks performed better but still relied on selective search. Faster R-CNN [18] was the first network introduced that performed detection without relying on selective search. Faster R-CNN used a CNN network known as region proposal network (RPN) [18] to find region proposals. In the year 2016, DenseNet [37] was introduced. DenseNet solved the vanishing-gradient problem and reduced the number of parameters required for training.

Later, Mask R-CNN [38] an extension of Faster R-CNN was introduced. Mask R-CNN extended Faster R-CNN [18] to pixel-level image segmentation by introducing an additional branch. Later in 2017, Retina-Net [39] was introduced, utilizing feature pyra-

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 28 April 2022 doi:10.20944/preprints202204.0279.v1



mid networks (FPN) [40] and focal loss to improve features and perform better detection. To solve the problem of overfitting, Cascade R-CNN [41] was introduced. The cascaded architecture reduces the Intersection Over Union (IoU) mismatches during training and inference time. Extending the network architecture of Cascade R-CNN, Hybrid Task Cascade [42] was introduced in 2019 with an additional branch for segmentation tasks. As backbones are an essential component of object detection algorithms, recently proposed several works have been continuously improving the results over the years. One such example is Swin Transformer [43] introduced recently in 2021. The transformer-based architecture allows for greater efficiency by introducing a window-based self-attention mechanism and Hierarchical feature maps generation.

#### 2.2.2. Challenging Environment

Recent advancement in deep learning-based algorithms has given rise to various approaches to improve object detection in challenging environments [11]. Sasagawa et al. [14] proposed an approach to detect objects under low illumination by taking advantage of state-of-the-art algorithms and techniques of transfer learning. The idea is to combine two models from different domains with the help of a generative model and glue layers. Further, to train both models properly, the authors propose using the knowledge distillation technique. First, spatial features are extracted from input by using an encoder-decoder network [44] composed of convolutional [45] and pooling layers [46]. With the help of pooling, layer features of different sizes and shapes are generated. The learned latent representation from the encoder-decoder network is propagated to the glue layer. After performing various experiments, the authors have established that the concatenation of all latent features produces the optimal result. After the glue layers, YOLO [47] is utilized to localize and identify objects. Another approach utilizing YOLO is proposed by Mate et al. [15] involving the use of thermal images instead of RGB images. As thermal images represent heat values, the authors establish that thermal images could improve object detection in low light environments and harsh weather conditions.

Another problem faced by object detection in a challenging environment is the loss of low-level features. Current object detection algorithms require high-level and low-level features to find objects and localize them [18]. The features help identify boundaries and different characteristics of objects present in the input image. These features are generally extracted from pre-trained backbones based on Feature Pyramid Network (FPN) [40]. To preserve low-level features, Yuxuan et al. [48] propose the fusion of contextual information in the backbone. The fusion of features helps in maximizing pre-trained channel information. The second problem faced by object detection algorithms is that when images captured in low light are passed through conventional hierarchical convolutions, the resulting output contains shallow rich features. Therefore, context fusion is incorporated in the backbone part of the network, thus preserving information in features. At every stage, low-level feature maps of the network are selected and fused with their successor. The resulting feature map is then provided to the network to detect objects.

Following the introduction of two-stage detectors in object detection algorithms and the ability of generative adversarial networks to learn image transformations, the combination of formal and latter has been used to improve object detection performance. One approach by Kun et al. [16] involves combining Deep Convolution Generative Adversarial Networks (DCGAN) [17] with Faster R-CNN [18] to detect objects in low light. The combination of DCGAN and Faster R-CNN involves three steps. First, DCGAN is used to learn and transfer the relationship between nighttime and daytime scenes. The second step is Multi-scale convolution feature fusion. Multi-scale convolutional feature fusion involves up-sampling and down-sampling of features to fuse them with their successors. The third step is to use an ROI pooling layer of different sizes to capture more detailed information. The authors argue that the standard ROI pooling layer reduces

 doi:10.20944/preprints202204.0279.v1



computational performance and loses the object's critical features. Finally, ROI pooling output is given to Faster R-CNN to obtain final results.

Another way of improving object detection is exploiting region-based convolutional neural networks like Mask R-CNN [38] and instance segmentation approaches [49,50]. Avramovic et al. [51] proposed a method that uses selective parts of the input image to detect traffic signs in an ardous environment. As the driver only focuses on particular positions like the front mirror and back mirror, the authors argue that object detection should only be applied to those regions instead of the whole image. Selective object detection is performed by selecting a limited amount of Regions of Interest (RoIs), thus reducing the computational cost. The authors have evaluated their approach using Mask R-CNN [38], and YOLO [47].

Kamal et al. [52] propose integrating two different network architectures based on Fully convolutional networks for semantic segmentation (FCNs) [53] to detect traffic signs. SegNet [54] and U-Net [55] are combined to detect signs, and a VGG-16 [56] based network is used for classifying detected signs to their corresponding classes. Segnet and U-Net are trained by extracting corners of images and using them as training data. The resulting output of four patches is combined to create an output mask for the original image. The authors have also used the L1 constraint term to modify Tversky Loss [57] to increase the detection of small traffic signs.

In a challenging environment, generic object detectors predict multiple bounding boxes for a single object. Most of the generated bounding boxes have low confidence and can be removed with a non-maximum suppression technique [58], but not all overlapping detections are removed. To address this, Eran et al. [59] propose a Soft-IOU layer using Jackkard distance as a quality detector between the predicted bounding box and the ground truth. The second step of the proposed solution is to treat predictions from the network as a clustering problem. A custom EM-Merger layer groups similar predictions into a single detection, thus removing overlapping detections. The authors have performed various experiments on the SKU-110K dataset using Retina-Net [39].

Apart from object detection algorithms, Semantic Image Segmentation (SIS) [60] have also been exploited to identify objects in arduous conditions. Unlike object detection algorithms, SIS tries to classify each pixel. Similarly, Ghose et al. [61] proposes a combination of saliency maps with thermal images to detect pedestrians in poor lighting conditions. Instead of using RGB and thermal images, the authors suggested that it is better to combine saliency maps and thermal images to find objects. First, thermal images are augmented with their corresponding saliency maps and then provided to deep saliency networks. The combination helps illuminate salient parts of the image while preserving textural information, making it easier for the network to find objects.

Similar to previous approaches of combining thermal images with RGB images, Zhengzheng et al. [62] propose fusing RGB images with thermal images to detect objects in adverse conditions. A two-stream convolution neural network architecture generates features from RGB and thermal images. The output is fused to form a single feature representation. The authors argue that the fusion of features from RGB and thermal images helps preserve mid-level features, which are necessary for refining object details. A Pyramid Pooling Module and a feature aggregation module to sharpen the object details are applied to the resulting features. The second contribution by the authors is the use of Convolutional Block Attention Module(CBAM) [63] to remove noise from features. CBAM is applied channel and spatial-wise. Finally, an average pooling layer is used to aggregate spatial information from features, and object detection is performed on them. The authors have used a combination of edge and cross-entropy loss to train the proposed architecture.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 28 April 2022 doi:10.20944/preprints202204.0279.v1



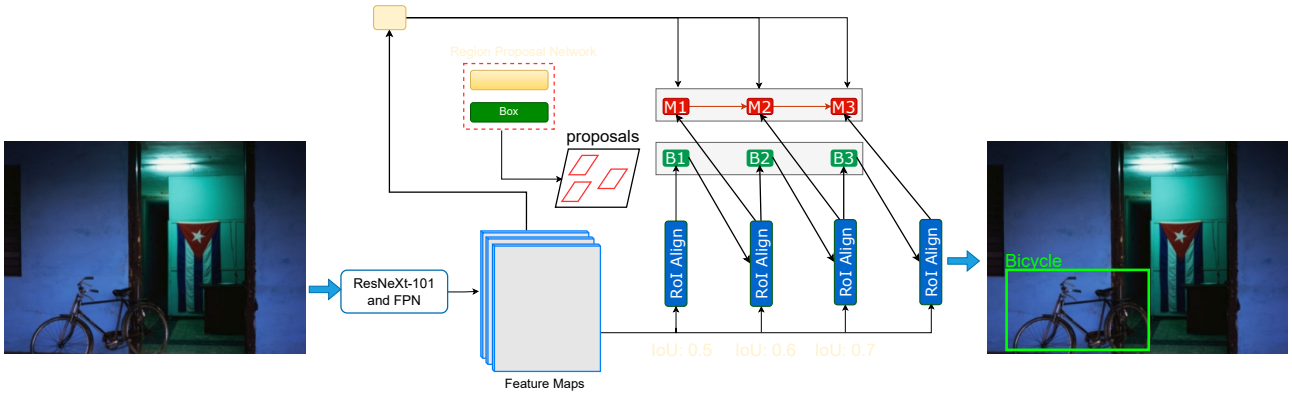



**Figure 2.** Illustration of the proposed framework. The combination of ResNext-101 and Feature Pyramid Network (FPN) extracts the spatial features from the input image on various scales. The features are propagated to the region proposal network and a fusion of semantic features and cascaded detection heads to perform final classification and localization.

## 3. Methods

### *3.1. Hybrid Task Cascade*

Cascading has been used in computer vision for a long time [64].It is a generic and dependable architecture that aids in improving performance. As a result, this design is employed to improve object detection performance. Iterative bounding box refinement [65] is a primitive approach for implementing cascading in object detection. There is an improvement in the performance of object detection. However, the improvement is not significant. Therefore, in object detection networks, a hybrid task cascade network presents a novel way of implementing the cascading design paradigm. To offer the spatial context, it first uses a fully convolutional branch.Second, it combines the detection and segmentation task within the cascade structure, allowing us to conduct both detection and segmentation at each level .As a result, we can name it collaborative multistage processing. Object detection and segmentation improve each other due to this cooperative multistage processing. Consequently, better detection can aid to enhance the performance of mask prediction and segmentation [42]. Figure 2 illustrates the proposed pipeline equipped with hybrid task cascade.

For bounding box prediction and segmentation, there are numerous heads. Furthermore, the data is processed at various scales. At the first step, the RPN provides input to the first bounding box head (B1), after which the cascade begins, with each consecutive bounding box head receiving input from the corresponding ROI align. Each mask head, however, receives two inputs. The semantic feature maps provide the first input. The ROI pooling provides the second input. Mask prediction combines the two to produce accurate masks. In summary, RPN generates the first object proposals that are processed by ROI pooling. The initial bounding box coordinates are generated by the head B1 using the ROI pooling output. It forecasts the object proposal's confidence also. In the second stage, M1 generates pixel-wise predictions in terms of masks. The other cascade levels follow the same pattern.

### *3.2. Backbone Network*

The backbone network is the fundamental part of the two-stage object detection methods since it extracts the spatial features and propagate the feature maps to the subsequent modules. In this paper, we utilize ResNeXt-101 [66] as the backbone network. The ResNeXt network extends the ResNet [67] architectures by providing the special cardinal features. A single layer of ResNeXt contains input channels, filter size, and output channels. This ResNeXt network has residual blocks. These residual blocks have two points: (i) The value of hyperparameters depends on spatial map size. (ii) If the spatial map size is reduced by 2, block-width becomes double. This provides uniform computation complexity.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 28 April 2022 doi:10.20944/preprints202204.0279.v1



In a neural network, neurons have aggregated-transformation in the form of inner product:

$$\sum_{j=1}^{C} w_j n_j \tag{1}$$

Where $n$ is an input vector fed to the neurons having C-channels while $w_j$ is the weight of filter for j-th channel. The ResNeXt [66] also includes this type of transformation in more specified form as short network. The aggregated transformation equation is as given below:

$$f(e) = \sum_{k=1}^{N} \tau_k(e) \tag{2}$$

Where $\tau_k(e)$ can be a temporal function to place $e$ into lower-dimension and transform it where N is the transformation size. The parameter N in Equation 2 is the same as C in Equation 1. However, these parameters are subject to change and can be tuned. The residual function can be mathematically explained as follows:

$$Y_{out} = e + \sum_{i=1}^{D} \tau_i(e) \tag{3}$$

Where $Y_{out}$ is the output function to be provided to Feature Pyramid and Region Proposal Network (RPN) of the employed HTC.

### *3.3. Feature Pyramid Network*

After the backbone network, the second component of two-stage detectors is a Feature Pyramid Network (FPN) [40]. FPN is a feature extractor that takes a single-scale image of arbitrary size as input and outputs different sized feature maps at multiple levels in a fully convolutional fashion. The feature pyramid generated helps object detection network by providing features at different scales. FPN is usually applied after backbone operation and is independent of it. The bottom-up pathway is a feed-forward computation of a backbone consisting of features maps at several scales. The advantage of building a feature pyramid network is generating stable features captured at different scales from higher pyramid levels. The features are enhanced with features from the bottom-up pathway via lateral connections.

### *3.4. Region Proposal Network*

Region Proposal Network (RPN) was introduced in Faster R-CNN. Once features are generated from the Feature Pyramid or backbone network, the next step in a two-stage object detection network is to find the regions where the objects can exist. RPN can predict regions where objects can exist instead of looking at every pixel, thus reducing the computational cost. Before RPN can predict possible candidate regions, anchors are drawn. Anchors are bounding boxes drawn with various sizes and scales on feature maps and represent the objects that networks need to detect. The size and shape of anchors can be configured from the dataset.

RPN network is composed of CNN layers and has a classifier and a regressor. The classifier part determines the probability of a proposal having the target object, and the regressor part regresses the coordinates of the proposal. RPN operates like any other CNN network by sliding a window over the features and predicting whether the anchors drawn in the region contain an object or not. Only the anchors with the highest IoU are assigned labels and used in later stages. RPN is trained along with other components of two-stage detectors during training. The loss function of RPN network is illustrated in Equation 4 as:

$$L(p_i, t_i) = (1/N_{cls}) * \sum L_{cls}(p_i, p_i^*) + (\gamma/N_{reg}) * \sum p_i^* L_{reg}(t_j, tj^*) \tag{4}$$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 28 April 2022 doi:10.20944/preprints202204.0279.v1



**Figure 3.** Samples taken from ExDARK dataset. The dataset has images captured in low light and indoor scenes.

where i donates the anchor index in a batch, and $p_i$ denotes the probability that an anchor is an object or not. Ground truth $p_i^*$ is one if the anchor is positive and is 0 if the anchor is negative. Similarly, $t_i$ denotes the vector of 4 parameterized coordinates of the predicted bounding box, and $t_i^*$ represents the ground truth box. The classification loss $L_{cls}$ is log loss over two classes (object vs non-object). For the regression $Lreg$, the loss function is shown in Equation 5 as:

$$L_{reg}(t_i, t_i^*) = R(ti - ti^*) \tag{5}$$

where R is robust loss function (smooth $L1$) defined in [35], $t_i$ represents ground truth box and $t_i^*$ represents predicted bounding box. The term $N_{cls}$ represents the normalization factor for classification loss and is equal to the batch size. The term $N_{reg}$ represents the normalization factor regression loss and is equal to the number of anchor locations. $\gamma$ is used for balancing parameters and by default is set to 10 unless stated otherwise.

## 4. Datasets

### *4.1. ExDark*

One of the most challenging and openly available datasets is the ExDARK [13] dataset created in 2020. The dataset comprises 7363 low-light pictures captured in different indoor and outdoor environments at nighttime. There is a total of 12 classes in the dataset. For the sake of variety, image enhancement techniques like de-hazing and blurring as augmentations are applied. The dataset contains the following classes: table, cat, people, motorbike, dog, cup, chair, bicycle, boat, bottle, bus, car, and cat. Figure 3 exhibits few samples from this dataset.

### *4.2. CURE-TSD*

CURE-TSD [68] is a large challenging dataset for the task of traffic sign detection. The dataset is composed of videos captured by driving a car around at different times of the day. Different augmentations like decolorization, blur, darkening, dirty lens, exposure, codex error, snow, and haze are applied to introduce variety. There are 14 types of traffic signs in this dataset: speed limit, goods vehicles, no overtaking, no stopping, no parking, stop, bicycle, hump, no left, no right, priority to, no entry, yield, parking. Figure 4 illustrates few samples of this dataset.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 28 April 2022 doi:10.20944/preprints202204.0279.v1



### 4.3. RESIDE

Another challenging dataset employed in our approach is RESIDE dataset [69]. The dataset is mainly for the task of object detection in difficult weather. The subset RTTS comprises 4,332 real-world hazy images representing different scenarios in a day. Images are collected manually through video cameras and annotated with bounding boxes localizing objects. The dataset contains various real-world occlusions such as hazy, rainy, and snowy weather. There are five annotated object classes in the dataset as bicycle, bus, motorbike, car, and person. Figure 5 depicts few samples from this dataset.

## 5. Experimental Results

### 5.1. Implementation Details

The codebase of the presented system is based on the MMdetection framework [70]. The backbone network is ResNext-101 which is pre-trained on ImageNet [45]. The cardinality of the backbone network is set to 64, and the bottleneck width is defined as four unless stated otherwise. We train on all three datasets with identical configurations. All datasets are fine-tuned for ten epochs, with a learning rate of 0.0025. SGD is used as an optimizer with a batch size of 4 on a single GPU machine. There are no augmentations applied during pre-processing, and only random horizontal flip is applied. Images sizes are kept variable in the range of 800 x 1388 while maintaining their aspect ratio.

### 5.2. Evaluation Protocol

As the problem of object detection in a challenging environment is identical to generic object detection, we evaluate our method by employing the similar evaluation metrics:

#### 5.2.1. Precision

Precision [71] defines as the percentage of a predicted region that belongs to the ground truth. The formula for precision is explained below:

**Figure 4.** Dataset samples taken from CURE-TSD. Heavy augmentation is applied to increase challenge for object detection algorithms.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 28 April 2022 doi:10.20944/preprints202204.0279.v1




**Figure 5.** Dataset sample taken from RESIDE. Dataset contains images captured in real challenging weather conditions.

$$\frac{\text{Predicted area in ground truth}}{\text{Total area of predicted region}} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{6}$$

where TP denotes True Positives and FP represents False positives.

#### 5.2.2. Recall

Recall, [71] is calculated as the percentage of the ground truth region that is present in the predicted region. The formula for the recall is given by:

$$\frac{\text{Ground truth area in predicted region}}{\text{Total area of ground truth region}} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{7}$$

Where TP is True Positives and FN represents False Negatives.

#### 5.2.3. Average Precision

Average Precision (AP) computes the average value of precision over different levels of recall. The higher the value of AP, the better performance and vice versa. The formula for calculating average precision is mathematically expressed as follows:

$$\text{AP} = \sum_{n} (R_n - R_{n-1}) P_n \tag{8}$$

where $R_n$ and $P_n$ are the precision and recall at the $n_{th}$ threshold.

#### 5.2.4. Intersection Over Union

Intersection Over Union (IOU) [72] is one of the most critical evaluation metrics that is regularly employed to determine the performance of object detection algorithms. It is the measure of how much the predicted region is overlapping with the actual ground truth region. IOU is defined as follows:

$$IOU = \frac{\text{Area of Overlap region}}{\text{Area of Union region}} \tag{9}$$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 28 April 2022 doi:10.20944/preprints202204.0279.v1



**Table 1.** Comparison between the proposed method and previous state-of-the-art results on ExDark dataset. $AP_s$ denotes average precision for small area, whereas $AP_m$ represents average precision for medium area and $AP_l$ depicts average precision for large area. The IoU threshold is also defined in the table. Best results are highlighted.

| Methods | mAP(0.50:0.95) | $AP^{50}$(0.50) | $AP_s$(0.50:0.95) | $AP_m$(0.50:0.95) | $AP_l$(0.50:0.95) |
|---|---|---|---|---|---|
| Ahmed et al. [11] | 0.67 | 0.93 | 0.50 | 0.61 | 0.71 |
| Yuxuan et al. [48] | 0.34 | 0.64 | 0.03 | 0.17 | 0.40 |
| Loh et al. [13] | 0.49 | 0.79 | - | - | 0.53 |
| Chen et al. [73] | 0.32 | - | - | - | - |
| Our Method | **0.71** | **0.94** | **0.57** | **0.69** | **0.75** |

#### 5.2.5. Mean Average Precision

Mean Average Precision (mAP) is another extensively applied evaluation metric for category-specific object detection. The mAP is the mean of average precision computed over all the classes. Mathematically, it is explained by:

$$\text{mAP} = \frac{1}{N}\sum_{i=1}^{N} AP_i \tag{10}$$

where $AP_i$ is the average precision for a given class explained in Section 5.2.3 and $N$ depicts the total number of classes.

### *5.3. Result and Discussion*

To assess the capabilities of the proposed method, we evaluate the proposed system on publicly available three challenging datasets. This section discusses the results achieved on all of three datasets.

#### 5.3.1. ExDark

We validate the performance of our system on the challenging ExDark dataset [13]. Table 1 presents the quantitative analysis of the proposed method. Moreover, it compares our results with previous state-of-the-art methods. Our method surpasses the previous state-of-the-art results with an mAP of 0.71 on a varying IoU threshold from 0.5–0.95. On the IoU threshold of 0.5, our method achieves an AP of 0.94.

The promising results on the low illuminance dataset illustrate that the extra segmentation module present in the employed HTC network facilitates the network to detect objects even in darker conditions. For complete understanding, Figure 6 depicts an instance of localizing and classifying a car in a dark image. Although the car is difficult to detect with a naked eye, our system detects it with a confidence of 100%.

**Comparison with State-of-the-art Methods**

By looking at Table 1, it is evident that our approach beats the prior best results with an mAP difference of $\tilde{4}$ points. The previous best results have been achieved by Ahmed et al. [11] with an mAP of 0.67, and Loh et al. [13] by achieving an mAP of 0.49.

#### 5.3.2. RESIDE

Analogous to ExDark, we report the performance on the RESIDE dataset, which is explained in Section 4.3. By analyzing Table 2, one can observe that the proposed method can further enhance the performance of object detection on the challenging RESIDE dataset. On an IoU threshold range from 0.5 to 0.95, we achieve an mAP of 0.52, whereas the AP of the proposed system goes to 0.81 on an IoU threshold of 0.5.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 28 April 2022 doi:10.20944/preprints202204.0279.v1




**(a)** Input Image **(b)** Predictions

**Figure 6.** Example of results achieved on the ExDark Dataset. Figure 6a represents an input image, whereas Figure 6b is the final output with the detected object. The blue color represents ground truth annotation, and orange is the network prediction.

Figure 7 exhibits the qualitative performance of the system. In Figure 7a, it can be seen that the image is visually challenging to interpret and Figure 7b shows the capabilities of the method to detect several objects present in the ground truth. However, on the left part of Figure 7b, one can observe a few instances of false positives with lower confidence scores.

**Comparison with State-of-the-art Methods**

As summarized in Table 2, the previous best results obtained on the RESIDE dataset are achieved by Ahmed et al. [11] with an mAP of 0.51. The proposed method in this paper pushes the previous state-of-the-art to the new best score of 0.52.

#### 5.3.3. CURE-TSD

CURE-TSD is the last dataset on which we assess the capabilities of the presented work. Table 3 presents the results achieved by our method on the CURE-TSD dataset. We achieve an mAP of 0.43 on an IoU threshold ranging from 0.5 to 0.95, whereas we attain an AP of 0.55 on an IoU threshold of 0.5. Furthermore, we achieve an AP of 0.06, 0.23, and 0.34 on the smaller, medium, and larger objects, respectively.

The qualitative analysis of our method is illustrated in Figure 4. In the mentioned figure, it can be perceived that the network has successfully detected a stop sign. However, owing to the high inter-class variance with other objects, the network produces a couple of false positives. Furthermore, the network produces a false positive by detecting a sign on the wall that appears similar to other objects in the dataset. This result raises an interesting question of how much prior context can improve this result [74].

**Comparison with State-of-the-art Methods**

By looking at Table 3, the previous best mAP attained on the CURE-TSD dataset is attained by Ahmed et al. [11] with an mAP of 0.28. However, the presented system outsmarts the prior results with an mAP of 0.43. Moreover, we observe a noticeable increase in the AP achieved on an IoU threshold of 0.5. It is essential to mention that Kamal et al. [52] achieved an AP of 0.94. However, we were unable to find the mAP score in the paper. Therefore, our results are not directly comparable with [52].

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 28 April 2022 doi:10.20944/preprints202204.0279.v1



**Table 2.** Comparison between the proposed method and previous state-of-the-art results on the RESIDE dataset. $AP_s$ denotes average precision for small area, whereas $AP_m$ represents average precision for medium area and $AP_l$ depicts average precision for large area. The IoU threshold is also defined in the table. Best results are highlighted.

| Methods | mAP(0.50:0.95) | $AP^{50}$(0.50) | $AP_s$(0.50:0.95) | $AP_m$(0.50:0.95) | $AP_l$(0.50:0.95) |
|---|---|---|---|---|---|
| Ahmed et al. [11] | 0.51 | 0.79 | **0.40** | 0.11 | 0.57 |
| Our Method | **0.52** | **0.81** | 0.26 | **0.40** | **0.57** |

**(a)** Input Image **(b)** Predictions

**Figure 7.** Example of results achieved on the RESIDE Dataset. Figure 7a represents an input image, whereas Figure 7b is the final output with the detected object. The blue color represents ground truth annotation, and orange is the network prediction.

## 6. Conclusion and Future Work

This research proposes an end-to-end optimizable system for tackling the challenge of object recognition in low-light and difficult environments. The proposed approach utilizes a hybrid task cascade network to effectively exploit pixel-level information at different cascade levels. On the ExDark, RESIDE, and CURE-TSD datasets, we get mAPs of 0.71, 0.52, and 0.43, respectively, by evaluating the suggested technique on three different challenging datasets. Unlike prior efforts, the presented method achieves its outcomes without pre-processing techniques such as picture augmentation. In the future, we plan to apply the idea of exploiting pixel-level information on other challenging datasets [59,75,76]. Furthermore, an end-to-end trainable pixel-level enhancement and learning approach would be another interesting future direction.

**Author Contributions:** writing—original draft preparation, K.A.H. and M.Z.A.; writing—review and editing, K.A.H., M.Z.A., M.L.; supervision and project administration, A.P., D.S. All authors have read and agreed to the submitted version of the manuscript.

**Funding:** The work leading to this publication has been partially funded by the European project INFINITY under Grant Agreement ID 883293.

**Institutional Review Board Statement:** Not applicable.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 28 April 2022 doi:10.20944/preprints202204.0279.v1



**Table 3.** Comparison between the proposed method and previous state-of-the-art results on the CURE-TSD dataset. $AP_s$ denotes average precision for small area, whereas $AP_m$ represents average precision for medium area and $AP_l$ depicts average precision for large area. The IoU threshold is also defined in the table. Best results are highlighted.

| Methods | mAP(0.50:0.95) | $AP^{50}$(0.50) | $AP_s$(0.50:0.95) | $AP_m$(0.50:0.95) | $AP_l$(0.50:0.95) |
|---|---|---|---|---|---|
| Ahmed et al. [11] | 0.28 | 0.38 | 0.06 | 0.23 | 0.34 |
| Kamal et al. [52] | - | **0.94** | - | - | - |
| Our Method | **0.43** | 0.55 | **0.12** | **0.26** | **0.53** |

**(a)** Input Image **(b)** Predictions

**Figure 8.** Example of results achieved on the CURE-TSD Dataset. Figure 8a represents an input image, whereas Figure 8b is the final output with the detected object. The blue color represents ground truth annotation, and orange is the network prediction.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Acknowledgments:** Not applicable.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Dai, J.; He, K.; Sun, J. Instance-aware semantic segmentation via multi-task network cascades. Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 3150–3158.
2. Hariharan, B.; Arbeláez, P.; Girshick, R.; Malik, J. Hypercolumns for object segmentation and fine-grained localization. Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 447–456.
3. Hariharan, B.; Arbeláez, P.; Girshick, R.; Malik, J. Simultaneous detection and segmentation. European Conference on Computer Vision. Springer, 2014, pp. 297–312.
4. Alberti, C.; Ling, J.; Collins, M.; Reitter, D. Fusion of detected objects in text for visual question answering. *arXiv preprint arXiv:1908.05054* **2019**.
5. Xu, K.; Ba, J.; Kiros, R.; Cho, K.; Courville, A.; Salakhudinov, R.; Zemel, R.; Bengio, Y. Show, attend and tell: Neural image caption generation with visual attention. International conference on machine learning. PMLR, 2015, pp. 2048–2057.

6. Wu, Q.; Shen, C.; Wang, P.; Dick, A.; Van Den Hengel, A. Image captioning and visual question answering based on attributes and external knowledge. *IEEE transactions on pattern analysis and machine intelligence* **2017**, *40*, 1367–1381.
7. Kang, K.; Li, H.; Yan, J.; Zeng, X.; Yang, B.; Xiao, T.; Zhang, C.; Wang, Z.; Wang, R.; Wang, X.; others. T-cnn: Tubelets with convolutional neural networks for object detection from videos. *IEEE Transactions on Circuits and Systems for Video Technology* **2017**, *28*, 2896–2907.
8. Zhang, P.; Lan, C.; Zeng, W.; Xing, J.; Xue, J.; Zheng, N. Semantics-guided neural networks for efficient skeleton-based human action recognition. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 1112–1121.
9. Vaswani, N.; Chowdhury, A.R.; Chellappa, R. Activity recognition using the dynamics of the configuration of interacting objects. 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings. IEEE, 2003, Vol. 2, pp. II–633.
10. Motwani, T.S.; Mooney, R.J. Improving Video Activity Recognition using Object Recognition and Text Mining. ECAI. Citeseer, 2012, Vol. 1, p. 2.
11. Ahmed, M.; Hashmi, K.A.; Pagani, A.; Liwicki, M.; Stricker, D.; Afzal, M.Z. Survey and Performance Analysis of Deep Learning Based Object Detection in Challenging Environments. *Sensors* **2021**, *21*, 5116.
12. Lin, T.Y.; Maire, M.; Belongie, S.; Bourdev, L.; Girshick, R.; Hays, J.; Perona, P.; Ramanan, D.; Zitnick, C.L.; Dollar, P. Microsoft COCO: common objects in context (2014). *arXiv preprint arXiv:1405.0312* **2019**.
13. Loh, Y.P.; Chan, C.S. Getting to know low-light images with the exclusively dark dataset. *Computer Vision and Image Understanding* **2019**, *178*, 30–42.
14. Sasagawa, Y.; Nagahara, H. Yolo in the dark-domain adaptation method for merging multiple models. European Conference on Computer Vision. Springer, 2020, pp. 345–359.
15. Krišto, M.; Ivasic-Kos, M.; Pobar, M. Thermal Object Detection in Difficult Weather Conditions Using YOLO. *IEEE Access* **2020**, *8*, 125459–125476.
16. Wang, K.; Liu, M.Z. Object Recognition at Night Scene Based on DCGAN and Faster R-CNN. *IEEE Access* **2020**, *8*, 193168–193182.
17. Gulrajani, I.; Ahmed, F.; Arjovsky, M.; Dumoulin, V.; Courville, A. Improved training of wasserstein gans. *arXiv preprint arXiv:1704.00028* **2017**.
18. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *arXiv preprint arXiv:1506.01497* **2015**.
19. Zou, Z.; Shi, Z.; Guo, Y.; Ye, J. Object detection in 20 years: A survey. *arXiv preprint arXiv:1905.05055* **2019**.
20. Viola, P.; Jones, M. Rapid object detection using a boosted cascade of simple features. Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001. Ieee, 2001, Vol. 1, pp. I–I.
21. Dalal, N.; Triggs, B. Histograms of oriented gradients for human detection. 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05). Ieee, 2005, Vol. 1, pp. 886–893.
22. Felzenszwalb, P.; McAllester, D.; Ramanan, D. A discriminatively trained, multiscale, deformable part model. 2008 IEEE conference on computer vision and pattern recognition. Ieee, 2008, pp. 1–8.
23. Agarwal, S.; Terrail, J.O.D.; Jurie, F. Recent advances in object detection in the age of deep convolutional neural networks. *arXiv preprint arXiv:1809.03193* **2018**.
24. Huang, J.; Rathod, V.; Sun, C.; Zhu, M.; Korattikara, A.; Fathi, A.; Fischer, I.; Wojna, Z.; Song, Y.; Guadarrama, S.; others. Speed/accuracy trade-offs for modern convolutional object detectors. Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 7310–7311.
25. Grauman, K.; Leibe, B. Visual object recognition. *Synthesis lectures on artificial intelligence and machine learning* **2011**, *5*, 1–181.
26. Andreopoulos, A.; Tsotsos, J.K. 50 years of object recognition: Directions forward. *Computer vision and image understanding* **2013**, *117*, 827–891.
27. Everingham, M.; Van Gool, L.; Williams, C.K.; Winn, J.; Zisserman, A. The pascal visual object classes (voc) challenge. *International journal of computer vision* **2010**, *88*, 303–338.
28. Betke, M.; Makris, N.C. Fast object recognition in noisy images using simulated annealing. Proceedings of IEEE International Conference on Computer Vision. IEEE, 1995, pp. 523–530.
29. Yuille, A.L.; Hallinan, P.W.; Cohen, D.S. Feature extraction from faces using deformable templates. *International journal of computer vision* **1992**, *8*, 99–111.
30. Papageorgiou, C.P.; Oren, M.; Poggio, T. A general framework for object detection. Sixth International Conference on Computer Vision (IEEE Cat. No. 98CH36271). IEEE, 1998, pp. 555–562.
31. Tsukiyama, T.; Shirai, Y. Detection of the movements of persons from a sparse sequence of tv images. *Pattern Recognition* **1985**, *18*, 207–213.
32. Xiao, Y.; Tian, Z.; Yu, J.; Zhang, Y.; Liu, S.; Du, S.; Lan, X. A review of object detection based on deep learning. *Multimedia Tools and Applications* **2020**, *79*, 23729–23791.
33. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. Proceedings of the IEEE conference on computer vision and pattern recognition, 2014, pp. 580–587.
34. Uijlings, J.R.; Van De Sande, K.E.; Gevers, T.; Smeulders, A.W. Selective search for object recognition. *International journal of computer vision* **2013**, *104*, 154–171.
35. Girshick, R. Fast r-cnn. Proceedings of the IEEE international conference on computer vision, 2015, pp. 1440–1448.

36. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 1–9.
37. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely connected convolutional networks. Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 4700–4708.
38. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask r-cnn. Proceedings of the IEEE international conference on computer vision, 2017, pp. 2961–2969.
39. Jaeger, P.F.; Kohl, S.A.; Bickelhaupt, S.; Isensee, F.; Kuder, T.A.; Schlemmer, H.P.; Maier-Hein, K.H. Retina U-Net: Embarrassingly simple exploitation of segmentation supervision for medical object detection. Machine Learning for Health Workshop. PMLR, 2020, pp. 171–183.
40. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 2117–2125.
41. Cai, Z.; Vasconcelos, N. Cascade r-cnn: Delving into high quality object detection. Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 6154–6162.
42. Chen, K.; Pang, J.; Wang, J.; Xiong, Y.; Li, X.; Sun, S.; Feng, W.; Liu, Z.; Shi, J.; Ouyang, W.; others. Hybrid task cascade for instance segmentation. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 4974–4983.
43. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin transformer: Hierarchical vision transformer using shifted windows. *arXiv preprint arXiv:2103.14030* **2021**.
44. Sutskever, I.; Vinyals, O.; Le, Q.V. Sequence to sequence learning with neural networks. *arXiv preprint arXiv:1409.3215* **2014**.
45. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems* **2012**, *25*, 1097–1105.
46. He, K.; Zhang, X.; Ren, S.; Sun, J. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE transactions on pattern analysis and machine intelligence* **2015**, *37*, 1904–1916.
47. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 779–788.
48. Xiao, Y.; Jiang, A.; Ye, J.; Wang, M.W. Making of night vision: Object detection under low-illumination. *IEEE Access* **2020**, *8*, 123075–123086.
49. Kopelowitz, E.; Engelhard, G. Lung Nodules Detection and Segmentation Using 3D Mask-RCNN. *arXiv preprint arXiv:1907.07676* **2019**.
50. Zhang, Q.; Chang, X.; Bian, S.B. Vehicle-damage-detection segmentation algorithm based on improved mask RCNN. *IEEE Access* **2020**, *8*, 6997–7004.
51. Avramović, A.; Sluga, D.; Tabernik, D.; Skočaj, D.; Stojnić, V.; Ilc, N. Neural-Network-Based Traffic Sign Detection and Recognition in High-Definition Images Using Region Focusing and Parallelization. *IEEE Access* **2020**, *8*, 189855–189868.
52. Kamal, U.; Tonmoy, T.I.; Das, S.; Hasan, M.K. Automatic traffic sign detection and recognition using SegU-net and a modified tversky loss function with L1-constraint. *IEEE Transactions on Intelligent Transportation Systems* **2019**, *21*, 1467–1479.
53. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 3431–3440.
54. Badrinarayanan, V.; Kendall, A.; Cipolla, R. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence* **2017**, *39*, 2481–2495.
55. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. International Conference on Medical image computing and computer-assisted intervention. Springer, 2015, pp. 234–241.
56. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* **2014**.
57. Salehi, S.S.M.; Erdogmus, D.; Gholipour, A. Tversky loss function for image segmentation using 3D fully convolutional deep networks. International workshop on machine learning in medical imaging. Springer, 2017, pp. 379–387.
58. Hosang, J.; Benenson, R.; Schiele, B. Learning non-maximum suppression. Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 4507–4515.
59. Goldman, E.; Herzig, R.; Eisenschtat, A.; Goldberger, J.; Hassner, T. Precise detection in densely packed scenes. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 5227–5236.
60. Chen, L.C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A.L. Semantic image segmentation with deep convolutional nets and fully connected crfs. *arXiv preprint arXiv:1412.7062* **2014**.
61. Ghose, D.; Desai, S.M.; Bhattacharya, S.; Chakraborty, D.; Fiterau, M.; Rahman, T. Pedestrian detection in thermal images using saliency maps. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, 2019, pp. 0–0.
62. Tu, Z.; Ma, Y.; Li, Z.; Li, C.; Xu, J.; Liu, Y. RGBT salient object detection: A large-scale dataset and benchmark. *arXiv preprint arXiv:2007.03262* **2020**.
63. Woo, S.; Park, J.; Lee, J.Y.; Kweon, I.S. Cbam: Convolutional block attention module. Proceedings of the European conference on computer vision (ECCV), 2018, pp. 3–19.
64. Cai, Z.; Vasconcelos, N. Cascade r-cnn: Delving into high quality object detection. Proceedings of the IEEE conference on computer vision and pattern recognition, June 18–22, 2018; Salt Lake City, Utah, USA, pp. 6154–6162.

65. Cai, Z.; Vasconcelos, N. Cascade R-CNN: high quality object detection and instance segmentation. *IEEE transactions on pattern analysis and machine intelligence* **2019**.
66. Xie, S.; Girshick, R.; Dollar, P.; Tu, Z.; He, K. Aggregated Residual Transformations for Deep Neural Networks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Hawaii, United States, July 21–26, 2017.
67. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. Proceedings of the IEEE conference on computer vision and pattern recognition, Las Vegas, Nevada, United States, June 26 – July 1, 2016, pp. 770–778.
68. Temel, D.; Chen, M.H.; AlRegib, G. Traffic sign detection under challenging conditions: A deeper look into performance variations and spectral characteristics. *IEEE Transactions on Intelligent Transportation Systems* **2019**, *21*, 3663–3673.
69. Li, B.; Ren, W.; Fu, D.; Tao, D.; Feng, D.; Zeng, W.; Wang, Z. Benchmarking single-image dehazing and beyond. *IEEE Transactions on Image Processing* **2018**, *28*, 492–505.
70. Chen, K.; Wang, J.; Pang, J.; Cao, Y.; Xiong, Y.; Li, X.; Sun, S.; Feng, W.; Liu, Z.; Xu, J.; Zhang, Z.; Cheng, D.; Zhu, C.; Cheng, T.; Zhao, Q.; Li, B.; Lu, X.; Zhu, R.; Wu, Y.; Dai, J.; Wang, J.; Shi, J.; Ouyang, W.; Loy, C.C.; Lin, D. MMDetection: Open MMLab Detection Toolbox and Benchmark. *arXiv preprint arXiv:1906.07155* **2019**.
71. Powers, D.M. Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. *arXiv preprint arXiv:2010.16061* **2020**.
72. Blaschko, M.B.; Lampert, C.H. Learning to localize objects with structured output regression. European conference on computer vision. Springer, 2008, pp. 2–15.
73. Chen, W.; Shah, T. Exploring Low-light Object Detection Techniques. *arXiv preprint arXiv:2107.14382* **2021**.
74. Sindagi, V.A.; Oza, P.; Yasarla, R.; Patel, V.M. Prior-based domain adaptive object detection for hazy and rainy conditions. European Conference on Computer Vision. Springer, 2020, pp. 763–780.
75. Hwang, S.; Park, J.; Kim, N.; Choi, Y.; So Kweon, I. Multispectral pedestrian detection: Benchmark dataset and baseline. Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 1037–1045.
76. Krišto, M.; Ivašić-Kos, M. Thermal imaging dataset for person detection. 2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO). IEEE, 2019, pp. 1126–1131.