HOCHSCHULE
RAVENSBURG-WEINGARTEN
UNIVERSITY
OF APPLIED SCIENCES

Computer Vision

Faculty of Electrical Engineering and Embedded System

Scientific Project

Object Detection for Night Vision Surveillance

Prajakta Ghodake [32916]

Weingarten, 3rd July 2020

Supervisor: Prof. Dr. rer. nat. Stefan Elser

Ghodake, Prajakta
Object Detection for Night Vision Surveillance
Scientific Project, Computer Vision
Hochschule Ravensburg-Weingarten,Weingarten, July 2020
E-mail ID : prajakta.ghodake@hs-weingarten.de

Abstract

In this project, a Deep Neural Network is evaluated for the task of Object Detection using low luminance dataset. Existing techniques uses a deep learning architecture to detect multiples objects during night time. In this work, a existing protocol for object detection is followed and a DNN is evaluated using Exclusively Dark Image dataset. Since the images were not annotated in the Ex-Dark dataset, a custom dataset is created using this dataset. The training images in the customized dataset are annotated using LabelImg tool. The model is trained and tested using training and testing dataset respectively. The result for this project demonstrates that the losses occuring due to low brightness are minimalized. For the evaluation of this model IOU metric is used and different IOU thresholds are compared to calculate the Average Precision.

Contents

List of Figures

Acronyms

AP Average Precision.

CNN Convolutional Neural Network.

FN False Negative.

FP False positive.

IOU Intersection Over Union.

mAP Mean Average Precision.

RCNN Regional Convolutional Neural Network.

ROI Region of Interest.

RPN Regional Proposal Network.

TN True Negative.

TP True Positive.

1 Introduction

Low-light is an unavoidable element of our everyday atmosphere that significantly affects the efficiency of our vision. Research works on low-light has seen a stable development, particularly in the field of object detection. Some recent work shows improvement in object detection by using denoising method based on bilateral filtering and wavelet thresholding [KMH+17]. Since, object detection is much related with video analysis and image understanding, it has attracted much research attention in recent years. Traditional object detection approaches are based on handcrafted features and low trainable architectures, hence efficiency to detect object in dark light is quite low. It is more difficult at night-time to detect an object due to low contrast against the background. However, some proposed algorithms based on thermal images for multiple object detection which show a better result for object detection in low light condition [VRPU20]. In this work, a Neural Network is built that detects multiple objects at low contrast for night surveillance using Exclusively Dark Image dataset.

1.1 Faster R-CNN

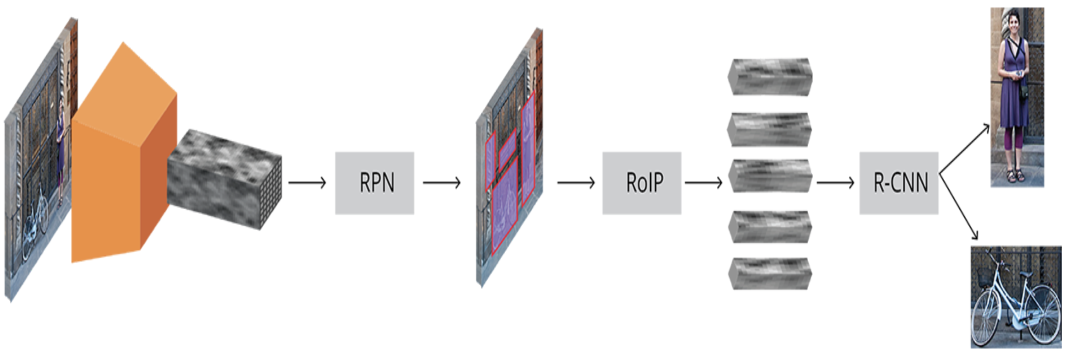


Figure 1.1: Complete Faster R-CNN architecture
[tl20]

R-CNN is the first step for Faster R-CNN architecture. It uses RPN to find the Region of Interest (RoI). In the figure 1.1, input image is passed through pre-trained CNN to get a convolutional feature map. Further, RPN is used for finding bounding boxes which contains images. These bounding boxes help RoI pooling and hence extracted features relevant to objects into a new tensor are used. Finally, the R-CNN uses this information to classify the information in bounding boxes [Bag20].

In this project, Faster R-CNN is used. As the classifier in this work is trained on a small custom dataset and since this architecture is very commonly used in the context of Transfer Learning, especially for training a classifier on a small dataset using the weights of a network trained on a larger dataset, FasterR-CNN was the better option for this work.

1.2 Exclusively Dark Image Dataset

For object detection research purpose predominantly in the low-light surroundings Exclusively Dark (Ex-Dark) image dataset is used [LC19a]. The Exclusively Dark (Ex-DARK) dataset has 7,363 low-light images from very low-light surroundings to dusk i.e. 10 different conditions circumstances with 12 object classes. For more information view : `https://github.com/cs-chan/Exclusively-Dark-Image-Dataset`

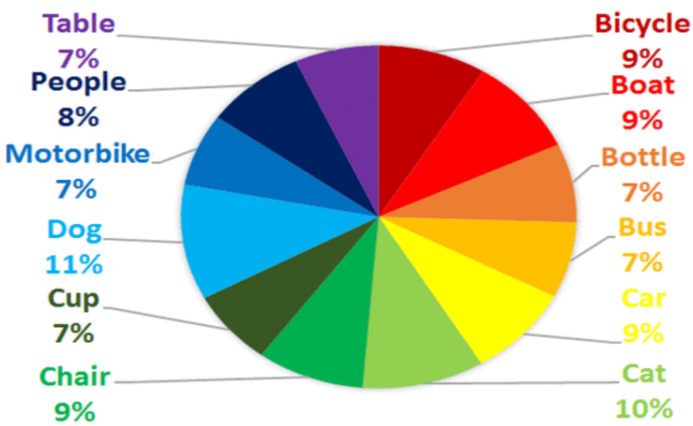


Figure 1.2: Class distribution in Exclusively Dark Image dataset [LC19b]

In the figure 1.2, we can see a class balance in this dataset that is there is no class skewing. In the figure 1.3, we can see different low luminance images belonging to a particular class with bounding boxes.

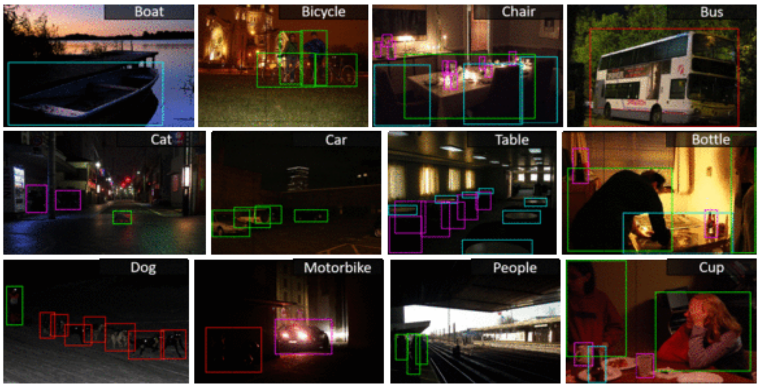


Figure 1.3: Images with low luminance
[LC19b]

1.3 Custom dataset

To evaluate the model a custom dataset is created using the Exclusively dark Image dataset. In addition, there was a need to have a high variety of image quality and even more classes with no class skewing. Also, the training dataset of Exclusively dark Image dataset was not annotated. Hence, there was a need to create a custom dataset. In this custom dataset, there are 11 classes and 816 images that are taken from Exclusively Dark Image dataset. This dataset is divided into training and testing samples. The training samples are annotated using LabelImg tool [dt16]. Label mapping is done in order to use this custom dataset. This includes the specification of the bounding boxes as well as the class names. Here is the link to this dataset: `https://mega.nz/folder/MqZl3QpD`.

1.4 TensorFlow

In this work, Tensor-flow API is used for object detection [Eva18]. The TensorFlow object detection API is the framework for creating a CNN that solves object detection problems. There are already pre-trained models in deep learning framework which are referred as Model Zoo. It comprises of collection of pre-trained models trained on the COCO dataset, the KITTI dataset, and the Open Images Dataset. These models can be used with these datasets for object detection and evaluation. Pre-trained models are also useful for initializing our own models when trained on the custom dataset. The various architectures are used in the pre-trained model like Faster-RCNN, RCNN, RPN etc.. In this project, ," faster_rcnn_inception_resnet_v1" model from the TensorFlow API is chosen. For more information view : `https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/detection_model_zoo.md`

2 Implementation

- **Task 1 : Installed Anaconda and setup for Object Detection directory structure and Anaconda Virtual Environment.**
  - In this task, the required environment for this project has been setup. For compiling module Anaconda Python IDE is used. All the required libraries for e.g. opencv, numpy, pillow etc. are installed, and necessary setup has been done.
- **Task 2: Collecting and labelling pictures.**
  - Data labelling is an essential step in a supervised machine learning task. In this work, bounding boxes image annotation type and Pascal VOC image annotation format is used to annotate the images. All the samples in the respective dataset (training, testing) are collected and labelled using LabelImg tool.
- **Task 3: Generation of training data.**
  - With the images labelled in task 2, TFrecords (.csv files) were generated that served as input data to the training model. Here, tensorflow Object detection API is used. With the help of TFrecords model is trained.
- **Task 4: Created a label map and configured training.**
  - Once model is trained next task was creating a label map which helped the trained model to classify what each object is, by defining a mapping of class names to class ID numbers.
- **Task 5: Testing and evaluating newly trained object detection classifier on a dataset.**
  - In this step, model is tested on unseen data i.e. testing dataset. The model is evaluated using IOU, Precision vs Recall curve and AP. The detection result i.e. AP is compared and analysed by setting two IOU threshold values.

3 Evaluation of Neural network

For evaluating an object detector there are multiple evaluation metric used for e.g. IOU, mAP, AP and so on. In this project, IOU is used as an evaluating metric.

3.1 IOU

Intersection over Union (IOU) [Pad18] is a measure based on Jaccard Index that evaluates the overlap between two bounding boxes. It requires a ground truth bounding box and a predicted bounding box. By applying the IOU one can predict if a detection is valid that is True Positive or is False Positive. Highest IOU is given by the overlapping area between the predicted bounding box and the ground truth bounding box divided by the area of union between them.

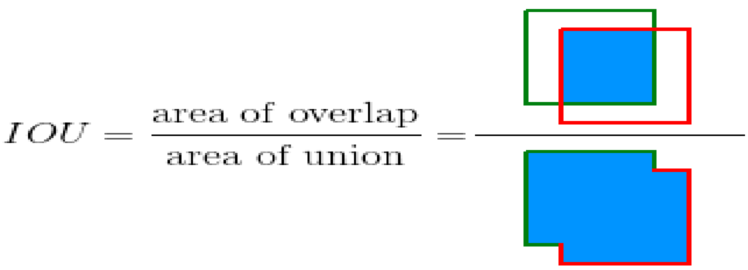


Figure 3.1: Intersection over Union
[Pad18]

In the figure 3.1, the IOU between a ground truth bounding box (in green) and a detected bounding box (in red) is given.

In this project, a threshold IOU of 0.7 and 0.5 is set and the IOU is calculated for every test image. Further, if the calculated IOU value is greater than or equal to the threshold value, it is considered that the object is detected correctly. And this is considered as a TP. In case if the calculated IOU is less than the threshold, it is considered as a FP. Additionally, a FN is considered as no detection of a ground truth and a TN would be all possible bounding boxes that are correctly not detected.

3.2 Precision and Recall

Precision and Recall is calculated with total number of TP, FP and FN. Precision refers to the percentage of the results that are relevant and recall refers to the percentage of all relevant results correctly detected by the algorithm [Pad18]. Precision and Recall formulae are as follows:

$$Precision = \frac{TP}{TP + FP} \quad \text{or} \quad Precision = \frac{TP}{all\ predictions}$$

$$Recall = \frac{TP}{TP + FN} \quad \text{or} \quad Recall = \frac{TP}{all\ groundtruths}$$

Considering the evaluation of a model w.r.t to each class, following are the interpretations:

- **HIGH RECALL + HIGH PRECISION :** The class is flawlessly handled by the model.
- **LOW RECALL + HIGH PRECISION :** The model not able to detect the class well but is highly trustable when it does.
- **HIGH RECALL + LOW PRECISION :** The model can detect the class, but the model also includes points of other classes in it.
- **LOW RECALL + LOW PRECISION :** The class is badly handled by the model.

It is not possible to maximize both these metrics at the same time, as one comes at the cost of another.

3.3 Precision vs Recall curve

For evaluating the performance of an object detector, Precision Recall curve is used. If the precision remains high as recall increases, the object detector for a particular class is considered good that means if the confidence threshold is changed, the precision and recall will still be high. Another way to identify a good object detector is to look for a detector that can identify only relevant objects (no False Positives = high precision), finding all ground truth objects (no False Negatives = high recall). A poor object detector needs to increase the number of detected objects (increasing False Positives = lower precision) in order to retrieve all ground truth objects (high recall). Hence, the Precision vs Recall curve usually start with high precision values, decreasing as recall increases [Pad18].

3.4 Average Precision

The AP is the area under the Precision Recall curve. It is calculated by plotting a recall and precision with increasing recall value. Than each recall level the precision value is replaced with maximum precision value to the right of that recall level. It is the precision averaged across all recall values between 0 and 1 [Pad18].

In this project, low AP is achieved due to the following reasons:

1. The dataset used is a Dark Image dataset and due to low luminance it was difficult to have more number of correct detections for some groudtruths i.e. TP the precision value is relatively low. Also, according to IOU thresholds values (0.5 and 0.7) the Precision and Recall values differed. Considering, IOU as 0.5, the AP is about 60% and IOU with 0.7, AP is about 51%.

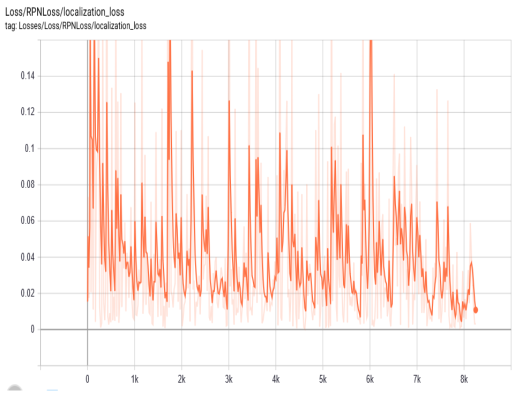
2. In this project, a reference dataset i.e. Exclusively-Dark image-dataset is used. Since, the training images in this dataset were not annotated, a custom dataset is created using Exclusively-Dark Image dataset. Since, labelling or annotation is a relatively time consumption task, there was incomplete labelling for some classes i.e. only 646 images are labelled and used to train the model. Having, less dataset for training, the calculated AP is low.

4 Result and Discussion

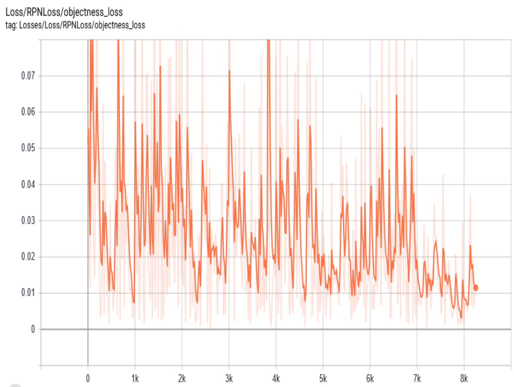
4.1 Evaluation of model

The following are the results obtained on training and testing the model.

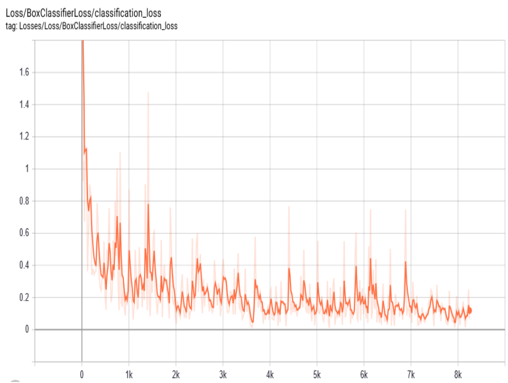
4.1.1 Training results



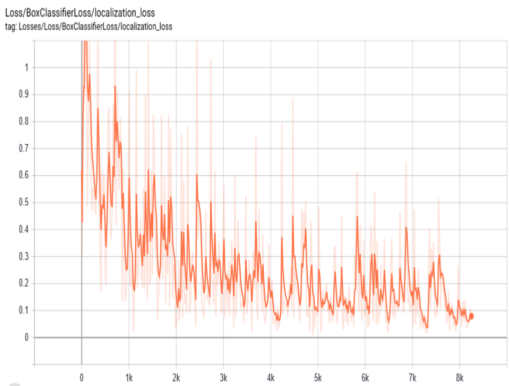
(a) RPN Loss / localizationloss



(b) RPN Loss / objectnessloss



(c) BoxClassifierLoss / classificationloss



(d) BoxClassifierLoss / localizationloss

Figure 4.1: Losses

In the Figure 4.1, all the training losses graphs are generated with tensorboard. The losses indicates the loss of information in the image during feature extraction.

1. The losses for RPN:
   - **RPN Loss / localization_loss:** Localization Loss or the Loss of the Bounding Box regressor for the RPN.
   - **RPN Loss / objectness_loss:** Loss of the Classifier that classifies if a bounding box is an object of interest or background.
2. The losses for the Final Classifier:
   - **BoxClassifierLoss / classification_loss:** BoxClassifierLoss / classification_loss: Loss for the classification of detected objects into different classes: People, Bus etc.
   - **BoxClassifierLoss / localization_loss:** Localization Loss or the Loss of the Bounding Box regressor.

4.1.2 Result for detection and classification of an input image

(a) Test image (b) Detected image

Figure 4.2: Detection and Classification result

The Figure 4.2 shows the object detection result. The yellow bounding boxes are the predicted boxes and red are the groundtruth boxes. It is clearly seen that the bottle in the image is correctly detected giving a TP and a confidence score of 84%. But, the people in the image are not detected due to incomplete labelling for 'Person' class and this counts as a FN.

4.1.3 Incomplete Labelling

The Figure 4.3 shows incomplete labelling for 'Person' class. Due to this incomplete labelling, the model was not trained properly for some specific classes for e.g. 'Person' in this project.

(a)

(b)

Figure 4.3: Incomplete labelling for 'Person' class

Figure 4.4: Object Detection result

In the Figure 4.4, the 'Person' is not detected due to the incomplete labelling of 'Person' class.

4.1.4 Precision vs Recall graphs

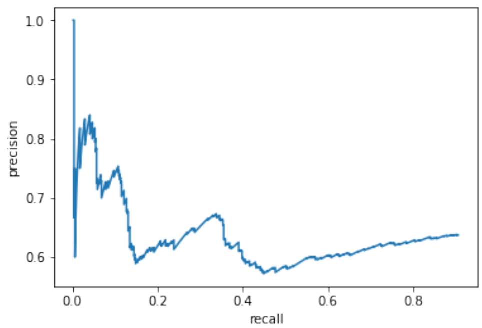


Figure 4.5: Precision vs Recall curve (IOU=0.5 , Recall average)

The Figure 4.5 gives the Precision Recall curve with IOU=0.5. The Recall value is averaged with the increase in the precision value from 0.6 to 1.

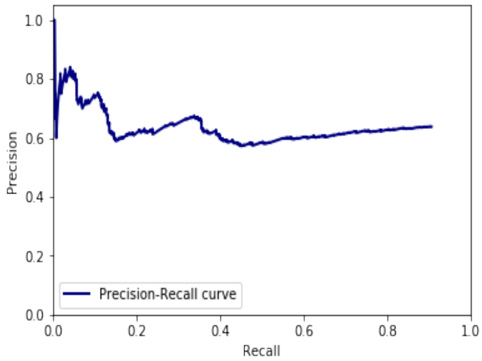


Figure 4.6: Precision vs Recall curve (IOU=0.5 , Precision average)

The Figure 4.6 gives the Precision Recall curve with IOU=0.5. The Precision value is averaged with the increase in the Recall value from 0 to 1.

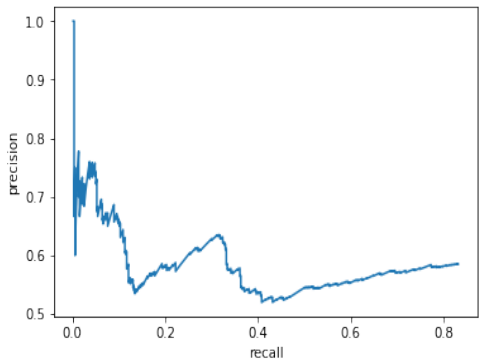


Figure 4.7: Precision vs Recall curve (IOU=0.7 , Recall average)

The Figure 4.7 gives the Precision Recall curve with IOU=0.7. The Recall value is averaged with the increase in the Precision value from 0.5 to 1.

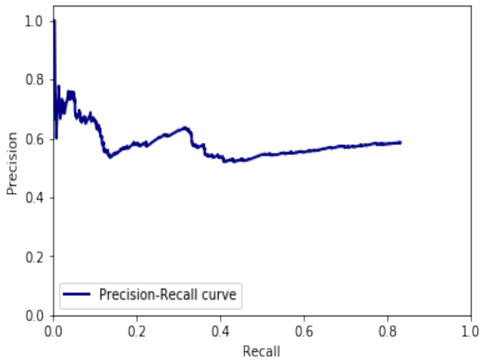


Figure 4.8: Precision vs Recall curve (IOU=0.7 , Precision average)

The Figure 4.8 gives the Precision Recall curve with IOU=0.7. The Precision value is averaged with the increase in the Recall value from 0 to 1.

4.1.5 AP results

1. By setting the IOU threshold value as 0.5, the AP obtained from the Precision Recall curve is 60%. If the prediction is perfect, IoU >=0.5, and if it completely missed, IoU < 0.5. A degree of overlap will produce a IoU value between those two and hence obtained the result of AP as 60%.

2. By setting the IOU threshold value as 0.7, the AP obtained from the Precision Recall curve is 51%. If the prediction is perfect, IoU >=0.7, and if it completely missed, IoU < 0.7. A degree of overlap will produce a IoU value between those two and hence obtained the result of AP as 51%.

3. In this project, comparing the two threshold values of IOU, we can conclude that, the AP for IOU=0.5 (60%) is better than IOU=0.7 (51%). This is because, considering a lower threshold of IOU there are more number of TP (correct predictions) and hence the calculated AP is better compared with higher threshold of IOU.

5 Conclusion and Future work

Object detection in minimum light conditions is an important part of visual surveillance since chances of unfortunate events are high in these situations. Night surveillance is a challenging task because of low brightness, low contrast and low appearance information. To tackle this issue, researchers have already developed varied algorithms.

This project was an attempt to follow the same protocol and aim to improve the accuracy of object detection using Neural Network in the night mode. In this work, it was possible to detect the objects in the images of different classes and could evaluate the model.

Comparing with the project proposal for this project, initially the training and testing dataset had to be the Exclusively Dark Image dataset. But later it was found out that the training samples were not labelled, so there was a need to create a custom dataset and label it. Due to time constrains, the labelling of more samples was difficult. For efficient and accurate performance, a network is implemented for object detection using Tensor-flow API and this model could detect an object with AP 60%.

In Future, to improve the detection and evaluation of any model using Dark Image dataset, image enhancement techniques like Linear contrast adjustment, contrast-limited adaptive histogram equalization (CLAHE) especially for dark images to increase the image quality and luminance level of the image can be used. Additionally, there should be more number of labelled samples so that the model is trained properly to have better detection and evaluation results.

In the field of Autonomous Driving, Object detection plays a vital role in order to have a safe drive especially in the low light conditions. In order to achieve safe and reliable autonomous driving, techniques of illumination compensation and illumination-invariant background subtraction should be used to solve the low-quality problem in object detection.

Bibliography

[Bag20] Saurabh Bagalkar: *Faster R-CNN: Using Region Proposal Network for Object Detection*, `https://medium.com/alegion/faster-r-cnn-using-region-proposal-network-for-object-detection-c113b6d`, 2020, [Online; accessed 17-June-2020].

[dt16] darrenl tzutalin: *LabelImg*, `https://github.com/tzutalin/labelImg`, 2016, [Online; accessed 1-June-2020].

[Eva18] Evan: *TensorFlow-Object-Detection-API-Tutorial-Train-Multiple-Objects-Windows-10*, `https://github.com/EdjeElectronics/TensorFlow-Object-Detection-API-Tutorial-Train-Multiple-Objects-Windows`, 2018, [Online; accessed 5-June-2020].

[KMH+17] Roman Kvyetnyy, Roman Maslii, Volodymyr Harmash, Ilona Bogach, Andrzej Kotyra, Żaklin Grądz, Aizhan Zhanpeisova und Nursanat Askarova: *Object detection in images with low light condition*, in *Photonics Applications in Astronomy, Communications, Industry, and High Energy Physics Experiments 2017*, Bd. 10445, S. 104450W, International Society for Optics and Photonics, 2017.

[LC19a] Yuen Peng Loh und Chee Seng Chan: *Getting to Know Low-light Images with The Exclusively Dark Dataset*, *Computer Vision and Image Understanding*, Bd. 178:S. 30–42, 2019.

[LC19b] Yuen Peng Loh und Chee Seng Chan: *Getting to Know Low-light Images with The Exclusively Dark Dataset*, `https://github.com/cs-chan/Exclusively-Dark-Image-Dataset/blob/master/Exdark.gif`, 2019, [Online; accessed 10-May-2020].

[Pad18] Rafael Padilla: *Object-Detection-Metrics*, `https://github.com/rafaelpadilla/object-detection-metrics`, 2018, [Online; accessed 12-June-2020].

[tl20] tryo labs: *Faster R-CNN: Down the rabbit hole of modern object detection*, `https://tryolabs.com/blog/2018/01/18/faster-r-cnn-down-the-rabbit-hole-of-modern-object-detection/`, 2020, [Online; accessed 17-June-2020].

[VRPU20] Pavan Sai Vemulapalli, Ajay Kumar Rachuri, Heena Patel und Kishor P Upla: *Multi-object Detection in Night Time*, *Asian Journal For Convergence In Technology (AJCT)*, Bd. 5(3):S. 01–07, 2020.