Semantic Segmentation in Low Light Disaster Scenes

Rob 10 Master Thesis

Cui Bo

Aalborg University
Department of Electronic Systems
Fredrik Bajers Vej 7B
DK-9220 Aalborg

Robotics
Aalborg University
http://www.robotics.aau.dk

AALBORG UNIVERSITY
STUDENT REPORT

Title:
Semantic Segmentation in Low Light Scenes

Theme:
Final Thesis

Project Period:
Spring Semester 2022

Project Group:
ROB10

Participant(s):
Bo Cui

Supervisor(s):
Letizia Marchegiani
Galadrielle Humblot-Renaux

Standard Page Count: 67

Date of Completion: 2nd June 2022

Abstract:

With the development of robotics, mobile robots have been gradually applied to various fields. In a disaster environment, mobile robots can complete tasks such as search and rescue more safely and efficiently. However, since the ambient brightness of the disaster scene is not always ideal, the robot not only needs to have the ability to distinguish different objects in the normal light scene, but also should be able to distinguish different objects in the dark environment. In this report, we achieve semantic segmentation of low-light scenes by combining a low-light image enhancement network and a semantic segmentation network. For low-light image enhancement, we restore low-light disaster scene images to normal-light images by training a DSLR[1] network. For semantic segmentation network, we propose modified PSPNet on the basis of PSPNet[2]. Experiments show that our modified PSPNet outperforms original PSPNet on the database we use. In addition, we also use modified PSPNet to verify that the method of low-light enhancement and then semantic segmentation can effectively improve the accuracy of low-light scene semantic segmentation. Furthermore, due to the lack of paired low-light disaster scene databases in this field, we also synthesize a low-light disaster scene database-LLDSD based on the PST900[3] dataset.

Preface

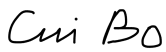
This project is created by 10th semester student of Robotics at Aalborg University.

The code for the project can be seen here:
https://github.com/Cuibo121381/low-light-disaster-scene-semantic-segmentation.

The citation style used in this report is Chicago Manual of Style; the citation inside of the sentence indicates it refers to the same sentence, whereas the citation placed after the dot mark indicates that it refers to the whole paragraph. The links are marked with blue text and are clickable. This report uses '.' as decimal point and ',' as thousands separators in numbers. The reading should be done in a chronological order.

Many thanks to my supervisors, for providing me guidance, feedback and inspiration during the entire project.

— ROB10, Aalborg University, 2nd June 2022



Bo Cui
<bcui20@student.aau.dk>

Glossary

LLDSD Low Light Disaster Scene Dataset.

LLIE Low Light Image Enhancement

mIoU Mean Intersection over Union, calculates the intersection ratio of ground truth and predicted segmentation.

PSNR PSNR is the ratio of the maximum power of the signal to the noise power, which measures the quality of the reconstructed image. The higher the PSNR index, the better the image quality.

SSIM SSIM is an indicator to measure the similarity of two images, and its value range is [0, 1]. The larger the value of SSIM, the smaller the degree of image distortion and the better the image quality.

Contents

1 - Introduction

With the development of robotics, when performing tasks such as search and rescue at disaster sites, mobile robots can be used to complete tasks more simply and safely[4].

Although robots can collect a lot of data (images, videos, point clouds, etc.), how to extract effective information from these data is still a challenging task. Especially in a disaster environment, the data collection mission for the robot is often challenging (collapsed walls, damaged items, etc.). The robot needs to understand these scenes and distinguish obstacles and targets in the scene, which requires the robot to have the ability of semantic segmentation.

Since the fully convolutional network (FCN)[5] and SegNet[6] were proposed, many semantic segmentation methods based on convolutional neural network (CNN) have been developed and applied in many fields, such as medicine[7], agriculture[8] and autonomous vehicles[5], etc.

These semantic segmentation methods exhibit high performance on different databases. However, most semantic segmentation studies mainly deal with normally exposed or brighter images[9], and the performance of these networks drops significantly when dealing with low-light images[10].

To solve this problem, there have been many image restoration works based on convolutional neural networks. CNNs trained using large-scale datasets can restore images affected by fog or haze into higher-quality images. For example, in work [11][12], it is shown that multi-scale CNN and deep convolutional generative adversarial networks have good results in recovering images affected by haze.

However, although many harsh environments (fog, haze) have become research hotspots, there are few studies on the work under extreme conditions such as disaster environments. One reason for the lack of relevant work in disaster conditions is the lack of suitable datasets. Due to safety and other factors, it is difficult to collect well-annotated images of the ground-truth in disaster areas after a disaster. Nonetheless, existing works [13][3][14] propose some very good disaster environment databases and efficient semantic segmentation methods.

But existing databases of disaster environments consist almost entirely of bright pictures. However, after a disaster, such as an earthquake, mine disaster, etc., the brightness of the environment may not be ideal. In low light conditions, the exposure time of the camera will be longer, which makes it more prone to movements of lens and blur during the shot. Also, since the camera's ISO is higher in low light, more noise is produced[10]. In this environment, existing methods suffer significant performance degradation because they

are not trained on low-brightness databases.

Considering these problems faced by the semantic segmentation of disaster scenes in low-light environments, we first propose a method to simulate low-light environment images, and based on the PST900[3] database, we synthesize a database LLDSD that can be used for low-light disaster scene semantic segmentation tasks. Then, we achieve semantic segmentation in low-light disaster scenarios by combining a low-light image enhancement network and a semantic segmentation network. We use DSLR[1] to enhance the database in LLDSD to convert low-light images to normal-light images. Then, we modify the original PSPNet[2] to enhance the semantic segmentation network performance. Different from the original PSPNet, we modify the loss function according to the characteristics of the labels in the used PST900 to improve the quality of semantic segmentation.

The arrangement of this paper is as follows. Chapter 2 presents previous methods for low-light database synthesis, low-light image enhancement, and semantic segmentation. Chapter 3 describes the question we define for this report. Chapter 4 describes the method we used in detail, and Chapter 5 describes and analyzes the experimental procedure and experimental results. Chapter 6 discusses our work.

2 - Related Research

2.1 Low Light Disaster Scene Dataset

2.1.1 Low Light Dataset

Low-light scenes are a very common phenomenon when taking pictures of low-light scenes. Low-light scenes can drastically reduce the quality of the image, causing more detail loss and low contrast, not only affecting subjective perception, but also the performance of many computer vision systems[15].

Several existing low-light datasets generally consist of pairs of low-light and normal-light images. The two main ways to construct low-light images are: taking multiple photos with different camera configurations or synthesizing low-light images from normal-light images. These datasets are presented in Table 2.1. R/S in the table represents whether the picture is Real or Synthetic.

Name	Number	Scene	R/S
LOL	500	I	R
SCIE	4413	I+O	R
VE-LOL-L	2500	I	R+S
AdobeFiveK	5000	I+O	R
SID	5094	I+O	R
DRV	202	I+O	R
SMOID	179	O	R

Table 2.1: Overview of public low-light datasets. The second column shows how many images are in the database. The column 'Scene' shows the scenes they were captured, 'I' means the images are taken indoor, and 'O' means outdoor.The last column shows whether the picture is Real or Synthetic. Among these databases, DRV[16] and SMOID[17] are vedio databases.

LOL

LOL[18] contains 500 pairs of RGB images captured in real scenes. Each low-light image has a paired normal light image. The images are captured by using different ISO and exposure time.

SCIE

SCIE[19] is a low light dataset containing 4413 multi-exposure images. It includes images of both indoor and outdoor scenes. Each normal-light image(589 in total) in the database has corresponding 3 to 18 low-contrast images with different exposures, so a total of 4413 multiple-exposure images are included, the images are stored as RGB format.

Adobe FiveK

MIT-Adobe FiveK[20] is also a common dataset for low-light enhancement. The database contains 5,000 images, all of them were manually retouched. The images in the database are all in RAW format, and [20] also provides a method that can process images into RGB format.

SID

SID[21] contains over 5000 low-light images in raw format,the corresponding high-light reference images are captured by long-exposure. Each low-light image in the database corresponds to a long-exposure image with different exposure times.

VE-LOL

VE-LOL[22] consists of two databases: VE-LOL-L and VE-LOL-H. The former one is a pairwise database used to train image enhancement networks, containing 2,500 pairs of images. Of these, 40% are synthetic. VE-LOL-H is a face detection dataset consists of over 10,000 unpaired images.

2.1.2 Disaster Scene Dataset

Table 2.2 presents some published datasets in the search and rescue(SAR) domain, all of these were captured in normal light conditions. Till now there are no Low Light datasets in SAR domain. The table also indicates the scene of the images (indoor (I), outdoor (O), and underground (U)), the total amount of data in the dataset, and whether the database is a real scene (R) or a synthetic scene (S).

DISC

DISC is a synthetic dataset of disaster scenes proposed by Jeon et al[14]. The database mainly simulates disaster scenarios in fifteen different locations. The dataset consists of

Name	Number	Scene	R/S
DISC	300K	I&O	S
UMA-SAR	77min	Outdoor	R
PST900	894	Underground	R
Petrıcek Dataset	10k	Indoor	R

Table 2.2: Overview of SAR Datasets. The second column shows how many images are in the database. The column 'Scene' shows the scenes they were captured.The last column shows whether the picture is Real or Synthetic

over 300K stereo image pairs, all of them have corresponding ground truth images.

UMA-SAR

The UMA-SAR dataset proposed by Morales et al[13] is a collection of multi modal raw video captured from manned all-terrain vehicles in a closed-loop path. The database contains data not only from RGB camera but also from 3D lidar ,IMU and GPS system.

PST900

The PST900 dataset was proposed by Shreyas et al[3]. The dataset contains 894 pairs of RGB-T images and 3395 pairs of RGB images. The dataset is annotated with the following four categories: fire extinguisher, backpack, hand drill, survivor. Ambient lighting or visibility is ensured through the use of high-intensity LED.

2.1.3 Summary

Among all the database mentioned in this section, the following phenomena were found. For all datasets in the Low Light Enhancement domain, they generally consist of indoor, outdoor or road images, which do not include disaster scenes. Meanwhile, for all SAR domain databases, the collected images or videos of disaster scenes are mostly composed of normal light pictures. Although the PST900 database contains some images with poor lighting conditions, most of the images are still normal light images.

It can be seen from this that, whether in the field of Low Light Dataset or Disaster Scene Dataset, there is no database that can satisfy the two conditions of low light and disaster scene at the same time. Therefore, the proposal of a SAR database composed of low-light images is very important for the task of semantic segmentation of low-light disaster scenes.

2.2 Low Light Image Synthesis

Although some effective low-light databases already exist, collecting paired normal-light and low-light images in real environments is still a challenging topic, especially in disaster scene. Indeed, in order to ensure that the paired images are exactly the same, we assume that everything in the images shouldn't greatly move or change. In this case, the desired images are obtained by different exposure strategies (eg: long exposure in dark conditions for normal light images or under exposure in normal light conditions for low light images). However, the environment may change (aftershocks/collapses) at any time during the acquisition of images in a disaster environment, making it difficult to collect paired low light datasets. Besides, if there are any survivors in the scene, they could also move. In this case, low-light images can be created more simply and efficiently through low-light image synthesis.

Several papers propose efficient low-light image synthesis methods. In paper[23],Lv et al. created low brightness images through random gamma adjustment. After that, simulated random noise to different image by adding Poison noise.

Feifan et al. used linear and gamma transformation to create low-light images[24]. Also used the Gaussian-Poisson mixed noise model to add noise to the image. In addition, they created ground truth images through contrast amplification (combine different photos to get rid of over-exposed).

Kin et al. also proposed a method in paper[25]. They created low-light noisy images by combining gamma adjustment and Gaussian noise.

Wang et al first used two different SLR cameras(Canon and Sony) to capture images[26]. Then they further collect about 15% of images from the internet by searching for the keywords "underexposure", "low light" and "backlight". Afterwards, they recruited experts to prepare modified reference images for each collected image.

In paper[27], Zhou et al. proposed a series of ways to create low-light datasets to train an architecture which they called LEDNet. They proposed a Synthesis Pipeline to create darken image with noise and defocus blur. Figure 2.1 shows their pipeline for creating low-light images.

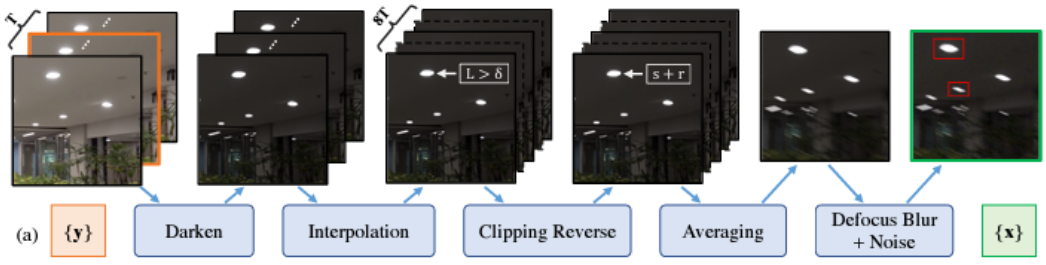


Figure 2.1: Image Synthesis Pipeline in LEDNet[27]

2.3 Image Enhancement

As mentioned in introduction, the performance of semantic segmentation networks on normal light images are better than on low light images[10]. Therefore, restore low light images to normal light images can be an feasible way to get better segmentation results. This section mainly introduces some classic low light enhancement methods.

2.3.1 Traditional Image Enhancement Method

Traditional methods can be mainly divided into two categories: method based on histogram equalization (HE) and method based on Retinex model.

Histogram Equalization

The core idea of histogram equalization method is to transform the histogram of the original image into a form of uniform distribution, thereby increasing the dynamic range of the image and achieving the effect of image enhancement. The results generated by conventional histogram equalization methods[28] often suffer from problems such as overexposure, loss of details, and color distortion. Therefore, a series of improved versions based on histogram equalization are designed to improve the above shortcomings.

For example, Kim et al.[29] developed a dual histogram equalization method, and Wang et al.[30] designed a binary sub-image histogram equalization method to achieve naturalization of exposure.

However, the existing methods based on distribution mapping still have phenomena such as color distortion that affect the look and feel of the enhanced results, one of the reasons is the lack of recognition and utilization of semantic information in the process of distribution mapping.

Retinex Method

Retinex theory[31] provides an intuitive physical description of the process of enhancing low-light images. The illuminance image is obtained by Gaussian filtering on the original image, and the illuminance image is obtained as accurately as possible, and finally the illuminance image is separated from the original image to obtain a reflection image (enhanced image).

Wang et al.[32] proposed a low-light image enhancement algorithm based on retinex theory. However, the images enhanced by this method often suffers from lack of details and insufficient brightness. Cai et al.[33]designed a joint intrinsic and extrinsic prior model for

optimizing illumination and reflections, but this approach tends to produce under-bright results.

With the deepening of the research, Guo et al.[34] constructed the first work that only considers modeling and solving for lighting, and the proposed method, named LIME, optimizes the resulting Initial lighting. But in most cases there will be overexposure.

In general this approach has some limitations: 1) The assumptions of the Retinex theory are too idealistic, this method will bring about loss of detail and distortion of color, 2) noise is usually ignored in the Retinex model, so the noise is preserved or amplified in the enhancement results, and 3) due to its complexity The optimization process has a relatively long running time.[35]

2.3.2 Image Enhancement Method Based on Neural Network

In recent years, with the continuous development of deep learning, low light enhancement methods based on deep learning has achieved remarkable success. Compared with traditional methods, deep learning-based image enhancement methods have better accuracy, robustness and faster speed[36].

Lore et al.[25] designed a low-light network (LLNet) deep auto encoder to improve the contrast of low-light images while taking into account denoising.

In addition, a series of works design the network structure based on the key theory of Retinex in the traditional model to give the algorithm the generalization ability on illuminance and reflection derived from the model.

Chen et al.[37] developed a Retinex-based low-light image enhancement network (RetinexNet). The network consists of two parts, namely the illumination estimation and the reflection layer estimation module. However, although this method can effectively improve the brightness of the image, some unknown artifacts and excessively refined details will cause image distortion.

The MBLLEN[23] algorithm proposed by Lv et al. is a multi-branch low-light image enhancement network. The core idea of this algorithm is that rich features in different levels can be extracted from the pictures, so image enhancement can be done through multiple sub-networks, and finally the output image can be generated through multi-branch fusion.

Further, Zhang et al.[38] proposed a simple and effective low-light image enhancement network (KinD). The network framework is similar to the RetinexNet architecture, but they changed the loss function and added an illumination adjustment layer, so that the light after enhancement is adjusted properly.

2.4 Semantic Segmentation

Semantic segmentation is a method that associates a label or category with each pixel of an image. It divides the image into regional blocks based on the semantic meaning and identifies the semantic category of each regional block. Finally, segmented images with pixel-wise semantic annotations are obtained.

Image semantic segmentation methods include traditional methods and CNN-based methods. With the development of deep learning, semantic segmentation technology is also gradually improving. The biggest difference between CNN-based semantic segmentation methods and traditional semantic segmentation methods is that the network can automatically learn the features of images and perform end-to-end classification learning[36]. Neural network-based semantic segmentation methods greatly improve the accuracy. In the following subsection, this report will give an overview of the most well established semantic segmentation architectures.

2.4.1 Fully Convolutional Networks(FCN)

FCN first feeds the image into CNN, obtains a series of feature maps through multiple convolutions and pooling, and then uses a deconvolution layer to upsample the feature map obtained by the last convolutional layer. Therefore, the size of the upsampled feature map is the same as the original image size[5]. Finally, the upsampled feature map is classified pixel by pixel, and the loss is calculated after passing through the softmax layer.

However, the limitation of the traditional FCN model is that it cannot perform fast, cannot effectively consider global context information,because the individual pixels are classified without adequate consideration of the pixel-to-pixel relationship[36].

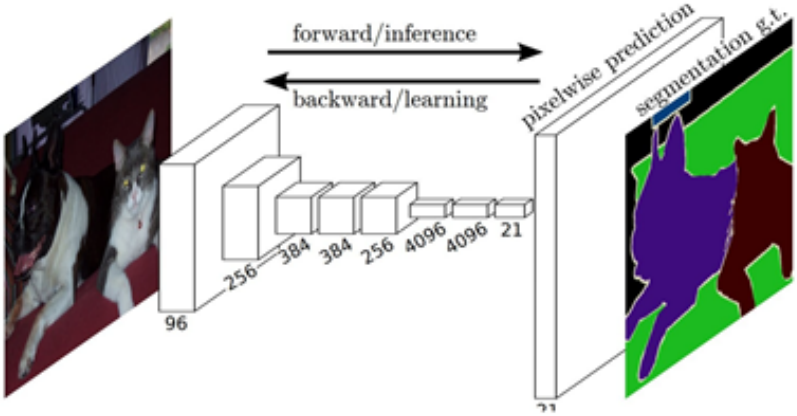


Figure 2.2: Structure for FCN[5]

2.4.2 SegNet

Badrinarayanan et al. proposed semantic segmentation network called SegNet[6]. SegNet uses an encoder-decoder structure in semantic segmentation. The encoder is mainly used to extract the features of the input image. The decoder upsamples the features until their dimensions match the input image. The output of the decoder is fed to a trainable softmax classifier to classify each individual pixel.

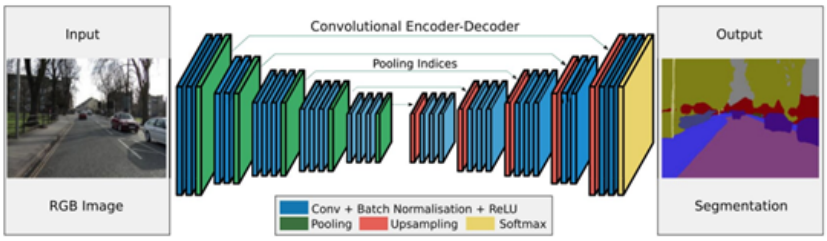


Figure 2.3: Structure for SegNet[6]

2.4.3 U-Net

Ronneberg et al. proposed a U-Net model for segmentation of biological microscope images[7]. U-net uses image tiles for training, so the amount of training data is much larger than the number of training images, which allows the network to gain robustness even with a small number of samples.

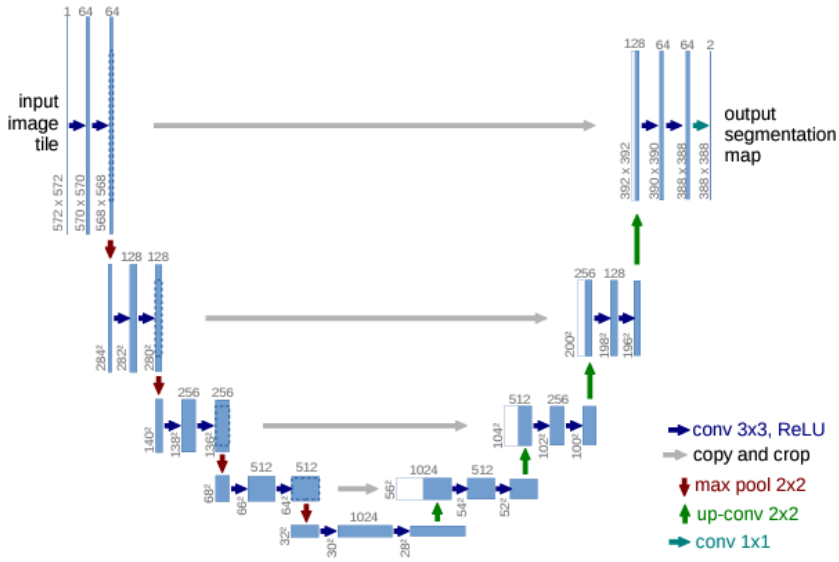


Figure 2.4: Structure for U-Net[7]

2.4.4 PSPNet

Zhao et al. proposed a Pyramid Scene Parsing Network (PSPNet)[39]. PSPNet first uses ResNet[40] to extract the features of the image. Then, the feature maps are parallel pooled using the pyramid pooling module to obtain four outputs of different sizes, which are then respectively upsampled and restored to the original feature map size.

Finally, it is connected with the previous feature map, and then the final predicted segmentation image is obtained through the convolution layer.

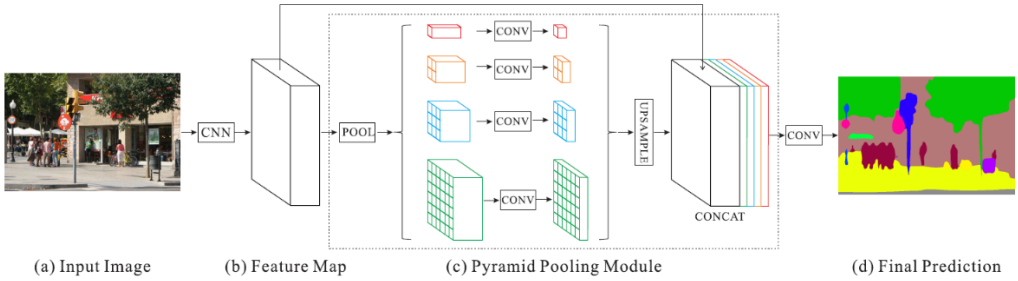


Figure 2.5: Structure for PSPNet[39]

2.4.5 Attention-Based Models

Chen et al. proposed an attention mechanism based semantic segmentation method[41]. They jointly train multi-scale images and attention models. The attention mechanism performs better than mean and max pooling, enabling the model to evaluate the importance of features at different scales.

2.4.6 Generative Models

Luc et al. proposed an adversarial training method for semantic segmentation[42].

As shown in Figure 2.7, the segmentation network on the left takes an RGB image as input and produces class predictions for each pixel. The adversarial network on the right takes the label map as input and generates class labels (1 for real labels, 0 for synthetic labels).

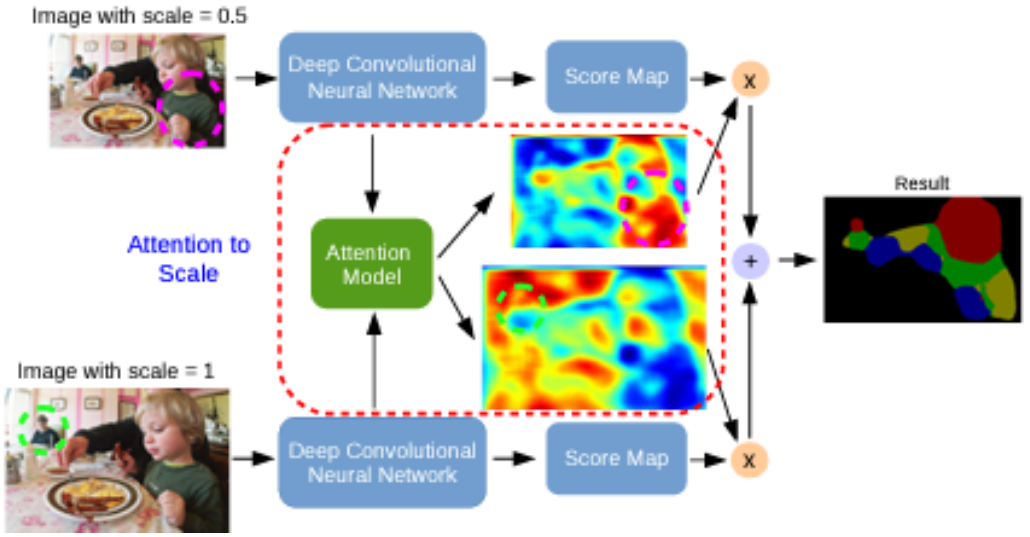


Figure 2.6: Structure for Attention-Based Model[41]

2.5 Low Light Scene Semantic Segmentation

According to the previous section, we can find that most of the existing deep learning based segmentation research proposes methods to segment various objects in daytime environment. However, in low-light environments, there is always lens shake due to the long exposure time and optical blur due to the movement of objects in the image, and low-light environments also produce more noise[10]. This makes the performance of the segmentation networks drops significantly[10].

To overcome these problems, many segmentation methods have been developed[43][44][45][46][47][48][49][50]. These semantic segmentation methods are mainly divided into two types, non-enhancement based segmentation method and enhancement based segmentation method. In studies [47][48][49], many different deep learning methods are used to improve the performance of low light segmentation. However, the visibility of nighttime images without enhancement applied is very low, making it difficult to train the segmentation network.

By performing image enhancement first and training the semantic segmentation network with the enhanced data set, the shortcomings of the above two methods are effectively avoided. Work [50] shows the performance of this approach and other transfer learning-based low-light image semantic segmentation algorithms. Therefore, this report adopts the same mode as [50], that is, the low-light image is enhanced first, and then the semantic segmentation network is trained through the enhanced image, so as to achieve the purpose of completing the low-light image semantic segmentation task.

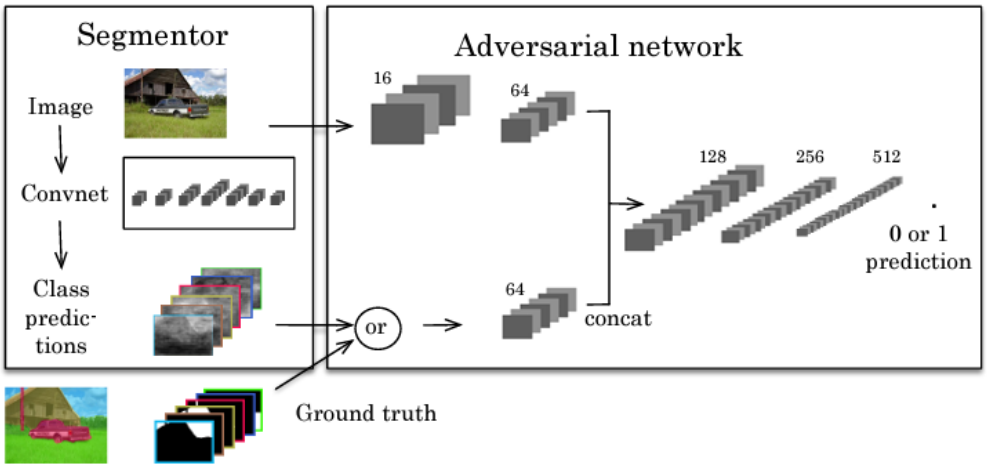


Figure 2.7: Structure for Segmentation based on GAN[42]

3 - Final Problem Formulation

From the introduction and the related research it can be seen that there is a lack of low light database in SAR domain. Besides, due to the low-light conditions, the features of objects may be blurred by various reasons, which greatly weakens the performance of many semantic segmentation networks in low-light environments.

Therefore, inside the different aspects of Semantic Segmentation in Low Light Scene the following problem statement is made:

- How to use a set of algorithms to make rescue robots have the ability of semantic segmentation in dark disaster environment?
  - How can the low light disaster scene database be created?
  - How to do semantic segmentation in a dark environment?
  - Is it possible to improve the accuracy of semantic segmentation by doing some processing on dark pictures compared to directly train a semantic segmentation network on the normal light dataset or on the low light dataset?

The following chapters will be focusing on answering this question and showcasing a possible solution with test results.

4 - Methods

In this report, to achieve semantic segmentation in low-light disaster environments, we formulate the following questions mention in Chapter 3. The outline of this report is as follows:
For *How can the low light disaster scene database be created?*, we will generate a synthetic low light disaster scene images dataset to make up for the lack of a database in the SAR domain.

For *How to do semantic segmentation in a dark environment?*, we propose a modified PSPNet to perform semantic segmentation on the images in our proposed database to verify the feasibility of semantic segmentation in low-light scenes.

For *Is it possible to improve the accuracy of semantic segmentation by doing some processing on dark pictures compared to directly train a semantic segmentation network on the normal light dataset or on the low light dataset?*, we enhance the images in our proposed database through a low-light image enhancement network. Afterwards, perform semantic segmentation on the enhanced image to test whether low light enhancement can improve the accuracy of semantic segmentation of low-light scenes.

4.1 Darkening and Noise Modelling

There are two main differences between low-light images and normal images: low brightness/contrast and the presence of noise[51].
The following subsections show some common methods when modelling dark and noise images.

4.1.1 Gamma Adjustment

Gamma transform is a nonlinear operation on the input image, so that the brightness of the output image has an exponential relationship with the brightness of the input image. The formula can be shown as:

$$I_{out} = A * I_{in}^{\gamma} \tag{4.1}$$

Gamma transform can be used for image enhancement, which improves shadow detail. To put it simply, through nonlinear transformation, the image is changed from a linear response of exposure intensity to a response that is closer to that of the human eye, that

is, an image that is too dark (underexposed) is corrected. Correspondingly, we can also darken a bright image in this way.

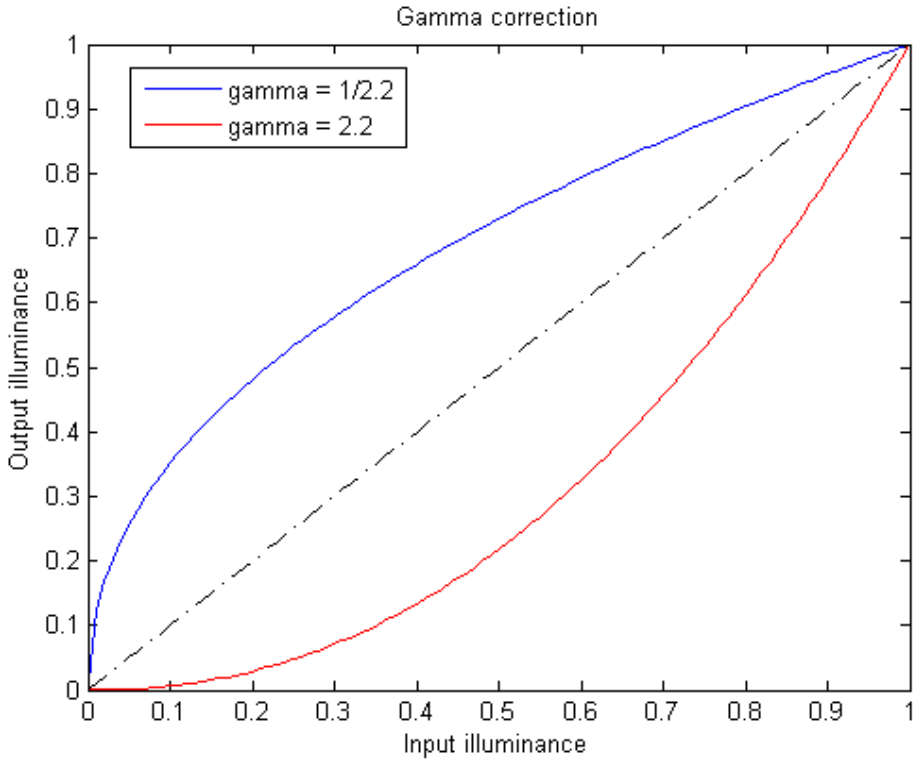


Figure 4.1: Figure of gamma correction. The blue and red curve shows how would the illuminance change by using gamma correction with different γ value.

The relationship between the input and output images after Gamma transformation is shown in the Figure 4.1. That can be summarized as follows:

- gamma>1, the brighter area is stretched, the darker area is compressed to be darker, and the overall image becomes darker;
- gamma<1, the brighter area is compressed, the darker area is stretched to be brighter, and the overall image becomes brighter;

4.1.2 Noise Model

The noise of low-light images may come from the initial image acquisition, quantization or subsequent image coding and compression transmission process[52]. The noise is mainly divided into three categories: 'Gaussian', 'Poisson', 'Salt&Pepper'. Among them,

salt & pepper noise is generally caused by interference in the environment (such as electromagnetic interference), internal timing errors of the sensor (ADC), etc[53]. Therefore, salt&pepper noise is generally not considered when simulating low-light pictures. Therefore, this report only considers Gaussian noise and Poisson noise when adding noise to images in the database.

Gaussian Noise

If the probability density function of the noise follows a Gaussian distribution, the noise is called Gaussian noise. Figure 4.2 shows the result of adding Gaussian noise to a gray-scale image.

Figure 4.2: The picture on the left[54] is a gray-scale image of the original image, and the picture on the right is a composite image with Gaussian noise added with $X \sim N(0, 1)$. Gray-scale images are used because it is easier to see the effect of noise on image quality.

Poisson Noise

Since CMOS has no way to collect all the photons emitted by the light source (the particle nature of light), the color of the photo will be skewed[55], which creates shot noise. At the same time, shot noise obeys Poisson distribution, so it is also called Poisson noise. Figure 4.3 shows how this noise will affect the original image.

Figure 4.3: The picture on the left[54] is a gray-scale image of the original image, and the picture on the right is a composite image with Poisson noise by Matlab function 'imnoise'. 'Imnoise' generates a Poisson noise for each pixel with mean equal to its pixel value. Gray-scale images are used because it is easier to see the effect of noise on image quality.

4.2 Semantic Segmentation

PSPNet is a classic semantic segmentation network, which has good multi-class segmentation accuracy[2]. Therefore, the semantic segmentation network used in this report is based on PSPNet. On the basis of original PSPNet, the network in this report is updated accordingly to adapt to the new database and application scenarios. The modified PSPNet structure is shown in Figure 4.4.

The overall structure of PSPNet can be summarized as: first extract features from input, then pyramid pooling, at last get prediction result.

4.2.1 Network Architecture

MobileNetV2[56]

Original PSPNet uses the Resnet series as the backbone feature extraction network in the report, but Resnet requires high computing power.

MobileNetV2[56] is a lightweight neural network structure produced by Google, which has two feature points: Inverted residuals and Linear bottlenecks. MobileNetV2 requires less memory and computation while maintaining accuracy.

Since the computing power of the laptop used in this report is limited, this report replaces the architecture of the feature extraction network with MobilenetV2[56].

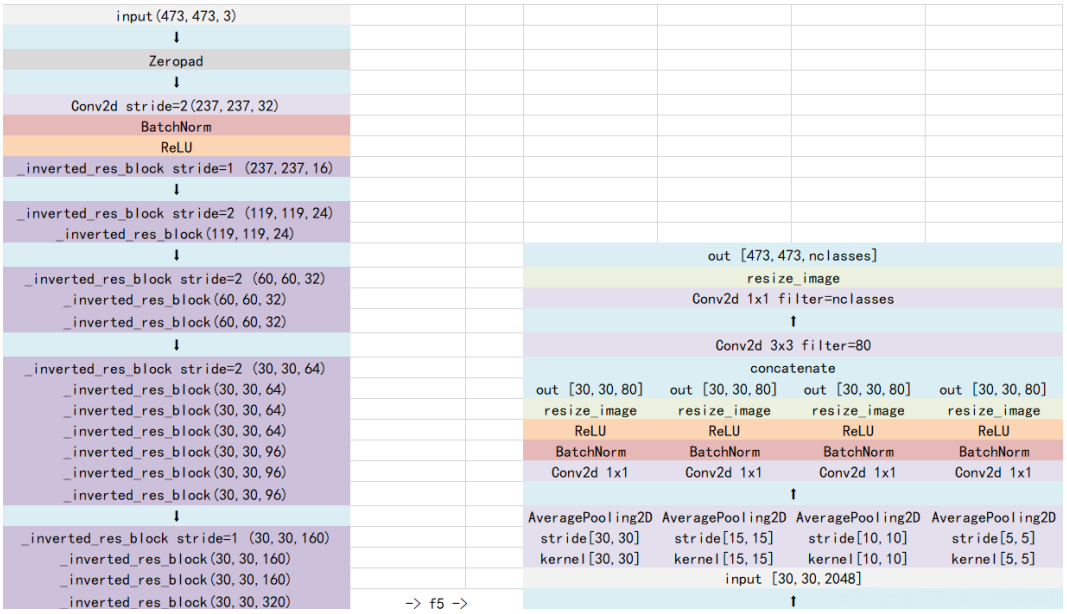


Figure 4.4: Whole Structure for PSPNet, the left part is the structure of the backbone 'MobileNetV2', the right part is the structure of Pyramid Pooling Module.

Figure 4.5 shows the network structure of MobileNetV2 used in this report.

Pyramid Pooling Module structure

The structure used to enhance feature extraction used by PSPNet is the Pyramid Pooling Module.
The approach of the PSP structure is to divide the acquired feature layer into regions of different sizes, and average pooling is performed within each region. Realize the aggregation of context information in different regions, thereby improving the ability to obtain global information.
In PSPNet, the PSP structure typically divides the input feature layer into 6x6, 3x3, 2x2, and 1x1 regions, and then performs average pooling within each region.
When the input feature layer of the PSP structure is 30x30x320, the specific composition of the PSP structure is as shown in Figure 4.4.

Use features to get predictions

Using the first two steps, we can obtain the features of the input image. At this time, we need to use the features to obtain the prediction results.

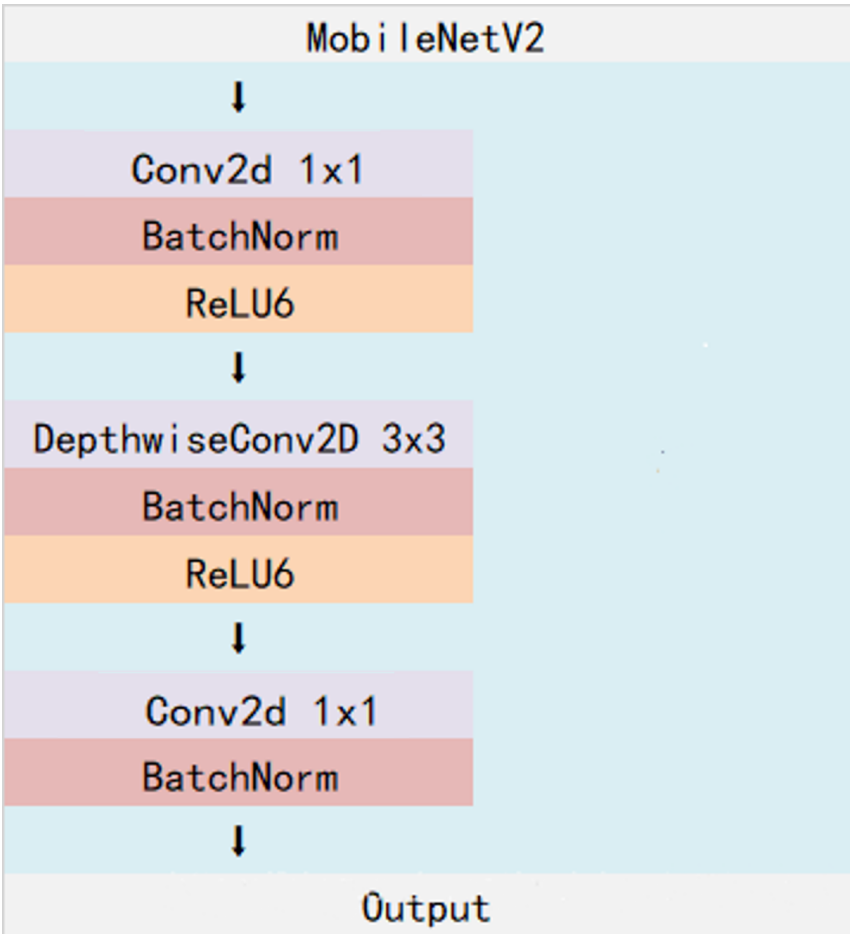


Figure 4.5: Structure for MobileNetV2[57]

The process of using features to obtain prediction results can be divided into three steps:

- Use a 3x3 convolution to integrate the features.
- Use a 1x1 convolution to adjust the channel and adjust it to Num_Classes.
- Up-sampling the concatenation result so that the final output layer has the same width and height as the input image.

4.2.2 Pre-training and Freeze Training

Pre-trained weights are used in this report to prevent the weights of the backbone from being too random. The weights are trained based on the cityscapes dataset in[2].

Due to the limitation of computing power used in this report, we chose to freeze training for the first 50 epochs. In the freezing phase, the backbone of PSPNet is frozen, and the feature extraction network does not change. It speeds up the training and only fine-tunes the network.

During the unfreezing phase, the backbone of the model is not frozen, and the feature extraction network changes, all the parameters of the network will be changed. In this report, both freeze training and unfreeze training are performed for 50 epochs.

4.2.3 Data Augmentation

Due to the small dataset used in this report, in order to prevent overfitting and improve the robustness and generalization ability of the network, data augmentation is performed on the images before inputting the training images into the network.

We first randomly scale the length and width of the input image and labels, and then flip the image (only when the random number < 0.5). At this time, if the length and width of the image is smaller than the length and width of the input image(473×473), we add black bars around the image to ensure the size of the image input to the network. If the length and width are larger than the original image, the enlarged image will be cropped to the input size by combining the crop operation.

In addition, we add operations such as Gaussian blur, rotation, and gamut transformation for further data augmentation. Parameters for data augmentation can be found in the code provided in this report.

4.2.4 Loss Function

In [2], PSPNet uses the cross-entropy function as the loss function. The paper innovatively calculates the loss after the residual block, and adds the result and the final loss of the network as the loss function. The loss function used in the report produces very desirable results.

This report makes some improvements to the loss function of original PSPNet. Due to the obvious difference in the size of objects in the PST900 data set (survivors are way bigger than hand-drills), there exists a problem of small objects, and there is also problem of imbalance between positive and negative samples, this report introduces a combination of two other loss functions to make PSPNet achieve better performance on PST900 dataset.

Cross Entropy Loss

Cross-entropy[58] checks each pixel one by one, comparing the predictions (probability distribution vectors) for each pixel class with our one-hot encoded label vector.It is widely

used for classification tasks and works well since semantic segmentation is pixel-level classification.

In multi-classification tasks, the soft-max activation function + cross entropy loss function is often used, because the output of the neural network is a vector, not in the form of a probability distribution. Therefore, the soft-max activation function is required to "normalize" a vector into the form of a probability distribution, and then use the cross-entropy loss function to calculate the loss.The formula for calculating the cross entropy loss is as follows:

$$L = -\sum_{i=0}^{C-1} y_i log(p_i) = -log(p_c) \tag{4.2}$$

Where $p = [p_0, ..., p_{C-1}]$ is a probability distribution, each element p_i represents the probability that the sample belongs to the i-th class; $y = [y_0, ..., y_{C-1}]$ is the one-hot representation of the sample label, when the sample belongs to the i-th class $y_i = 1$, otherwise $y_i = 0$; c is the sample label.

Dice Loss

The formula for Dice Loss is as follows[59]: in the formula K means the number of labels, N means the number of pixels of the label in the figure. The factor 2 in the numerator is because the denominator is double counting common elements between predict and ground truth.

$$L = 1 - \frac{1}{K}\sum_{j=1}^{K}\frac{2\sum_{i\in N} y_i^j \times p_i^j}{\sum_{i\in N} y_i^j + \sum_{i\in N} p_i^j} \tag{4.3}$$

In the image segmentation task, the cross entropy loss is to predict the class of each pixel, and then average all the pixels. In essence, it still performs equal learning on each pixel of the image, which leads to the fact that if there are imbalances in multiple categories on the image, the training of the model will be dominated by the most mainstream category. The network is more inclined to the learning of mainstream categories, and reduces the feature extraction ability of non-mainstream categories.

However, dice loss is calculated by dividing the intersection of prediction and GT by their total pixels, considering all pixels of a class as a whole. Moreover, the proportion of the intersection in the population is calculated, so it will not be affected by a large number of mainstream pixels and can have better performance.

Dice Loss has good performance for scenes with classes unbalanced problems[60]. But the training loss is easy to be unstable when dealing with small objects. This is because when using dice loss, generally when the positive sample is a small target, there will be serious oscillations, once the small target has some pixel prediction errors, it will lead to a large change in the loss value, resulting in a sharp change in the gradient.

To solve this problem, dice loss is always combined with cross entropy loss or focal loss[61]. In this report, we use Focal Loss to stabilize the training loss.

Focal Loss

He Kaiming's team introduced Focal Loss in the RetinaNet paper to solve the imbalance in the number of difficult and easy samples[62]. During the training process, the easy-to-learn labels can be easily predicted correctly. As long as the model classifies a large number of easy-to-learn labels correctly, the loss can be reduced a lot, resulting in the model not paying much attention to the hard-to-learn samples.

So the main idea of Focal loss is to find a way to make the model pay more attention to the samples that are difficult to learn. The expression of Focal Loss is as follows:p represents the probability that the predicted sample belongs to label c. When current sample is a positive sample, $y = 1$, otherwise $y = 0$. γ is used to let the network focus more on samples hard to classify. α is a parameter used to adjust the proportion of positive and negative samples.

$$L = \begin{cases} -\alpha(1-p_c)^{\gamma} log(p_c) & \text{if } y = 1 \\ -(1-\alpha)p_c^{\gamma} log(1-p_c) & \text{if } y = 0 \end{cases} \tag{4.4}$$

When γ is greater than 0, for positive samples, if it is a difficult-to-classify sample(when p is small), $(1-p)^{\gamma}$ will be very large, and the network will be more trained for difficult-to-train samples. The value of α ranges from 0 to 1. When $\alpha > 0.5$, the proportion of $y = 1$ can be relatively increased. This achieves a balance of positive and negative samples.

The experimental results of the paper shows when $\gamma = 2$, $\alpha = 0.75$, focal loss has better performance. Therefore, the parameters used in this report are the same as those recommended in the paper.

Since most of the labels in the database used in this report account for a small proportion of the entire image, ideally, the combination of Dice Loss and Focal Loss can make the network balance the learning between easy and hard samples and achieve the purpose of improving network performance. This report replaces the cross entropy loss of original PSPNet with the sum of Focal Loss and Dice Loss, that is:

$$Loss = FocalLoss + DiceLoss \tag{4.5}$$

In the following experiments, this report will respectively train the PSPNet using the cross-entropy loss function and the modified PSPNet using the loss function shown in Equation 4.5 on the PST900 dataset to test the effectiveness of the loss function used in this report.

4.3 Image Enhancement Method

In the previous chapter, this report introduced state-of-the-art low-light image enhancement methods. In a recent survey [36], we can compare the enhancement effect of several methods trained on the LOL dataset by this report, as shown in Figure 4.6. The input in the figure is a low-light image in the LOL dataset(the low light image is shown in Figure 4.7), and we can see that Zero-DCE, MBLLEN, and DSLR perform better than other

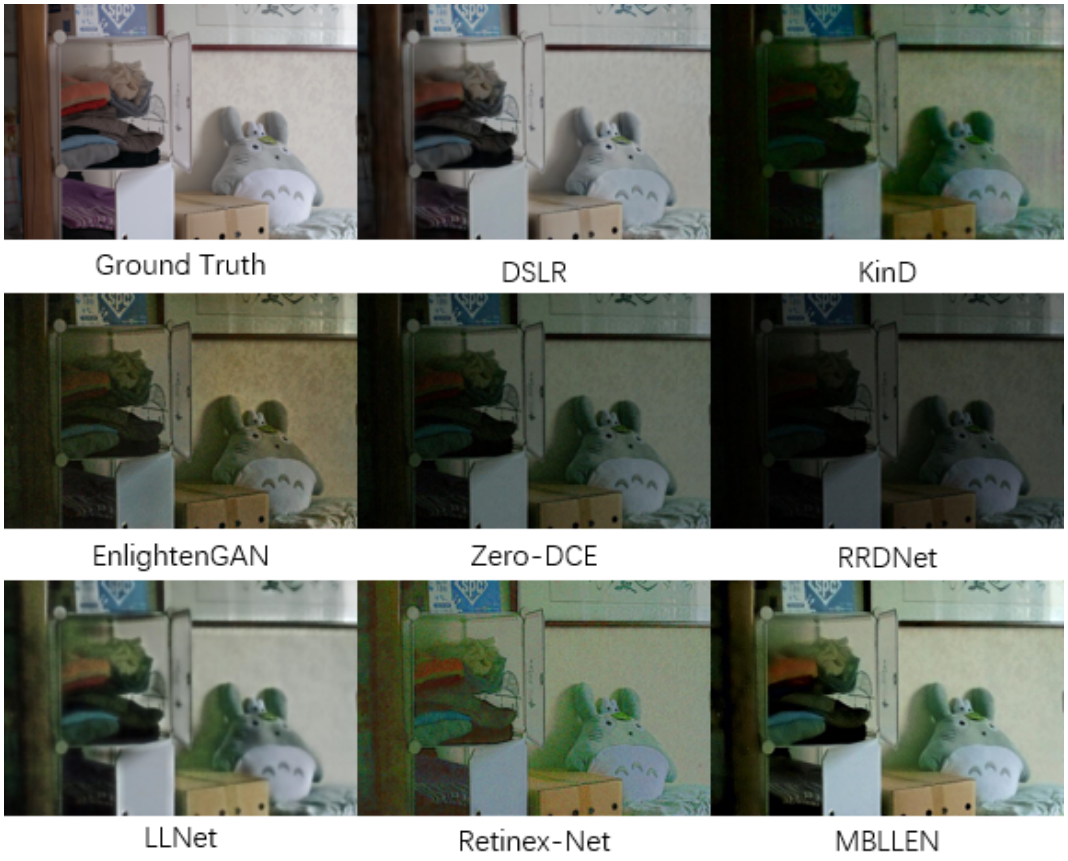


Figure 4.6: Enhancement Result of Different Low Light Enhancement. The ground truth image is provided by LOL dataset[18], the other images are the result of different image enhancement methods which is trained on LOL dataset[36].

methods in terms of brightness and color recovery, DSLR has the highest degree of color recovery. Among them, Zero-DCE doesn't need paired images for training, but to train a curve to adjust every pixel of the input image. In our report, we generated a dataset with paired low light and normal light images. Therefore, a supervised learning based method is more suitable for our report.

Since this report uses Pytorch to build a semantic segmentation network, in order to ensure the consistency of the compilation environment, the DSLR that provides the Pytorch version[1] is selected as the image enhancement network.

4.3.1 Network Structure

The main idea of DSLR[1] is to adjust global illumination and recover local details by exploiting Laplacian pyramids in image and feature space. The network framework can

[1]https://github.com/SeokjaeLIM/DSLR-release

Figure 4.7: Corresponding low light image in LOL[18]

be divided into three small encoder-decoder networks, each of which is used to learn the features of inputting different levels of the decomposition of the Laplacian pyramid.

The paper proposes a multiscale Laplacian-residual block (MSLB). Within each network, this module can decompose features into Laplacian pyramids, which will help restore contrast and brightness at image details. Each encoding-decoding module can be divided into three main parts, namely convolution block, deconvolution block and MSLBs.

At the end of the network, the enhanced results of the three sub-networks are combined into one image, which is the enhanced image.The structure of the network is shown in Figure 4.8.

4.3.2 Loss Function

The loss function used by the DSLR network mainly consists of three parts[1]: data loss L_d, Laplacian lossL_l and color lossL_c. The expression of the loss function is as follows:

$$L_T = \omega_1 L_d + \omega_2 L_l + \omega_3 L_c \tag{4.6}$$

In this report, ω_1,$\omega_2$and ω_3 are set to 2, 2 and 1 as recommended by the paper.

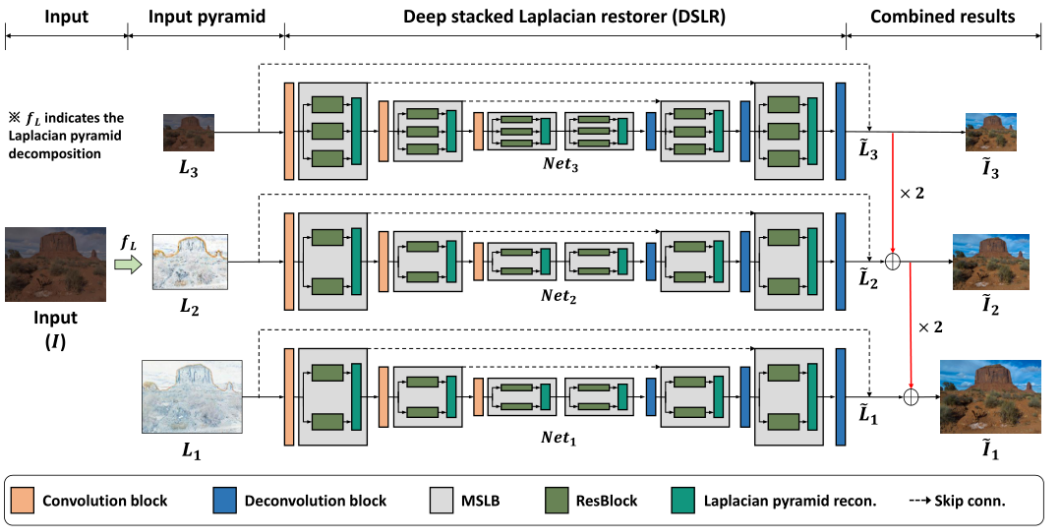


Figure 4.8: Overall Structure of DSLR[1]. In the figure, L1 refers to the input image, L2 and L3 refer to different levels of the decomposition of the Laplacian pyramid. At the end of the network, the red lines means the output of 3 sub-net will be combined as output.

Data Loss

Mean squared error is used as data loss in this report. The mean square error can judge the difference between the enhanced result and Ground Truth, and its expression is as follows:

$$L_d^k = \| I_k^{pred} - I_k^{real} \|^2 \tag{4.7}$$

$$L_d^{Total} = \frac{1}{T_k} \sum_{k=1}^{3} L_d^k \tag{4.8}$$

$L_d^k$is the mean squared error between the predicted value of each decision of the Laplacian pyramid and the current level of Ground Truth. Finally, the mean square errors of the three-layer network are added and normalized ($\frac{1}{T_k}$ is the normalization coefficient), and the result obtained is used as the data loss of the network.

Laplacian Loss

Due to the squared term in the formula for data loss, it often produces blurry restoration results[63]. In order to maintain the sharpness of the enhancement results, the mean absolute value error is used as the Laplacian loss, and the expression of the loss is as follows:

$$L_l^{Total} = \sum_{k=1}^{2} \frac{1}{T_k} | L_k - L_k^* | \tag{4.9}$$

L_k and L_k^* represent the predicted and true values of Laplacian residual image restoration at different levels, respectively. T_k represents the number of all pixels in the image.

Color Loss

Cosine similarity uses the cosine value of the angle between two vectors to measure the difference between two individuals. Compared with distance measures, cosine similarity pays more attention to the difference in direction of two vectors, rather than distance or length. Therefore, the cosine similarity loss function is used as the color loss to ensure that the enhanced color vector of each channel has the same direction as the corresponding ground-truth value. The color loss function is shown below:

$$L_c = \sum_{k=r,g,b} 1 - \frac{1}{T_k} \sum_{i \in T_k} \frac{\vec{x_{(i)}} \cdot \vec{y_{(i)}}}{\| x_{(i)} \|^2 \cdot \| y_{(i)} \|^2} \tag{4.10}$$

T_k represents the number of the pixels of the image, $\vec{x_{(i)}}$ represents the predicted position of each pixel color in each channel, and $\vec{y_{(i)}}$ represents the true position of each pixel color in each channel. When the predicted color is closer to the real color, the color loss will be closer to 0.

5 - Implementation and Result

The pipeline of this report is shown in Figure 5.1, as mentioned in last chapter, we will first generate a low light disaster scene dataset(yellow block in the figure). We will then enhance this dataset by DSLR[1], and will compare the result with several other enhancement methods(green block in the figure). We will also propose a modified PSPNet[2] which will be compared with original PSPNet. Then we will train the modified PSPNet with different database to check the performance of our method in low light disaster scene semantic segmentation task.

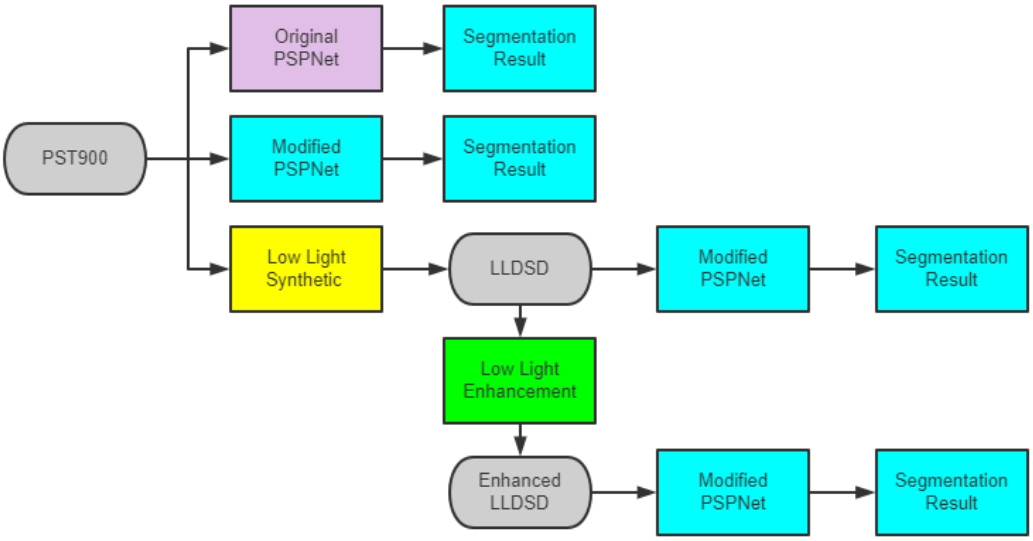


Figure 5.1: Pipeline of whole system in this report. The yellow block represents the process of generating Low Light Disaster Scene Dataset(LLDSD). The green block represents low light enhancement by using DSLR. Blue blocks shows the process of semantic segmentation on different dataset.

5.1 Low Light Disaster Scene Dataset (LLDSD)

According to the discussion in previous section, there is no specialized dataset for low light disaster scene, therefore, a low-light simulation method is proposed to synthesize low-light images from normal-light images. The aim is to provide the lighting conditions needed for our method and other further studies. Note that in the context of this thesis,

synthetic images refer to real images which have been synthetically degraded.
Many previous works use synthetic data as an effective alternative[64] to real data in different vision tasks. Therefore, this report generates a synthetic low-light image dataset by using the already annotated open source dataset - PST900[3]. The dataset construction pipeline is shown in Figure 5.2.

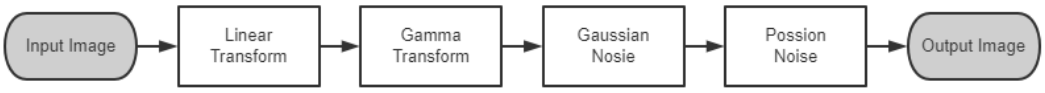


Figure 5.2: Pipeline of Image Synthesis. The first two step is to generate low light image. The following two part is to synthesize noise which is normal in real images captured in low light scene.

5.1.1 Target Image Synthesis

The low-light image generation process in this report is mainly divided into two parts: first, the brightness and contrast of the normal image are reduced by a transformation. The resulting image can be viewed as a low-light image without noise. After that, noise is added to the resulting low-light image. This produces an adequate number of paired low/normal light images which are needed for training of learning-based methods.

Low Light Image Synthesis

For noise-free low-light image generation, we apply a random gamma and linear transform to each channel of the normal image to generate low-light images, similar to [65].The low-light image simulation pipeline (without additional noise) can be formulated as:

$$I_{out} \quad = \quad B * (A * I_{in}^{\gamma}) \tag{5.1}$$

The part in parentheses represents the gamma correction of the image. A and B stands for linear transformation. In this report, when constructing each image, the three parameters is randomly sampled from uniform distribution: $A \sim U(0.9, 1)$, $B \sim U(0.5, 1)$ and $\gamma \sim U(3, 5)$. $A \sim U(0.9, 1)$ and $B \sim U(0.5, 1)$ are commonly used in the papers doing the same work [22][19]. For γ, some paper use $(1, 5)$, but in order to generate darker images, in this report we chose $(3, 5)$.

Figure 5.3 shows the image after Low Light Synthesis. We can see that the picture has darkened significantly and the contrast has dropped accordingly.

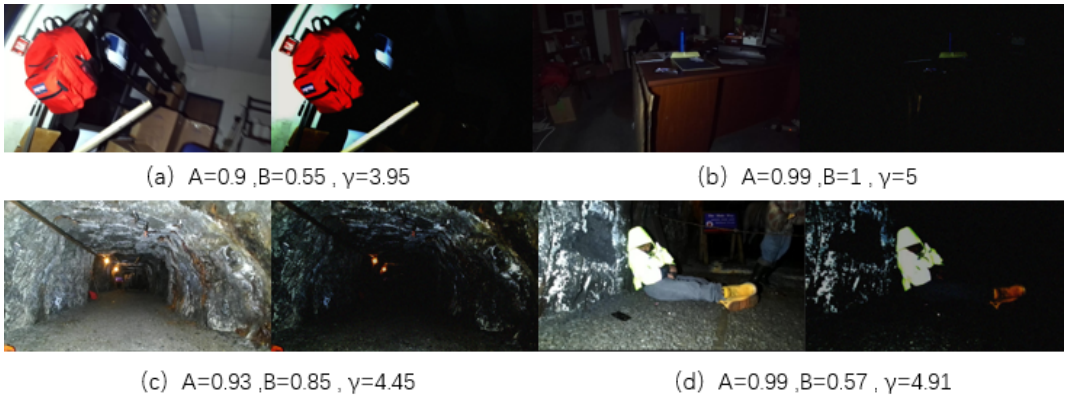


Figure 5.3: Several Pairs of Original Image(Left) and Darken Image(Right). When $\gamma = 5$, we can get the darkest image. The influence of how these 3 parameters will influence the darken result can also be seen in this image, the parameters are below each pair of images.

Noise Image Synthesis

Another important difference between low-light images and normal light is that low-light images have more noise. Therefore, this report mainly considers Gaussian noise and Poisson noise when creating the low-light database. In particular, this report refers to the Gaussian-Poisson mixed noise model used by[66], adding randomly generated Gaussian noise and Poisson noise to each image in the database. The noise model can be formulated as:

$$I_{out} = P(f(L + n(I_{in}))) \tag{5.2}$$

$$L = f^{-1}(I_{in}) \tag{5.3}$$

where $P(X)$ represents Poisson noise with added variance σ_p^2 and $n(X)$ is modeled as Gaussian noise with noise variance σ_g^2. $f(x)$ represents the camera response function. L is to generate irradiance image from input image. In this report, we followed the recommendation in [66], both σ_p^2 and σ_g^2 obey the uniform distribution of $U(0.003, 0.01)$. In addition, unlike in [66], the influence of CRF on the results is also ignored(the value of CRF is set to 0). This is because in report[67], the RGB images were taken by a stereo camera but they didn't show the exact CRF of this camera.

In Figure 5.4, we can see that there is a lot of noise in the image compared to before the noise was added.

In addition, it is shown in [23] that enhancement networks trained with multiple scenes with different brightness will perform better. Therefore, this report also generates two additional databases of the same size according to the same method as above, the only difference between them is the choice of gamma value. The gamma values of these three databases are $\gamma \sim U(0,2)$,$\gamma \sim U(2,3)$and$\gamma \sim U(3,5)$. The ranges are chosen to cover different light condition. So the dataset size used to train the DSLR network is 3359*3.

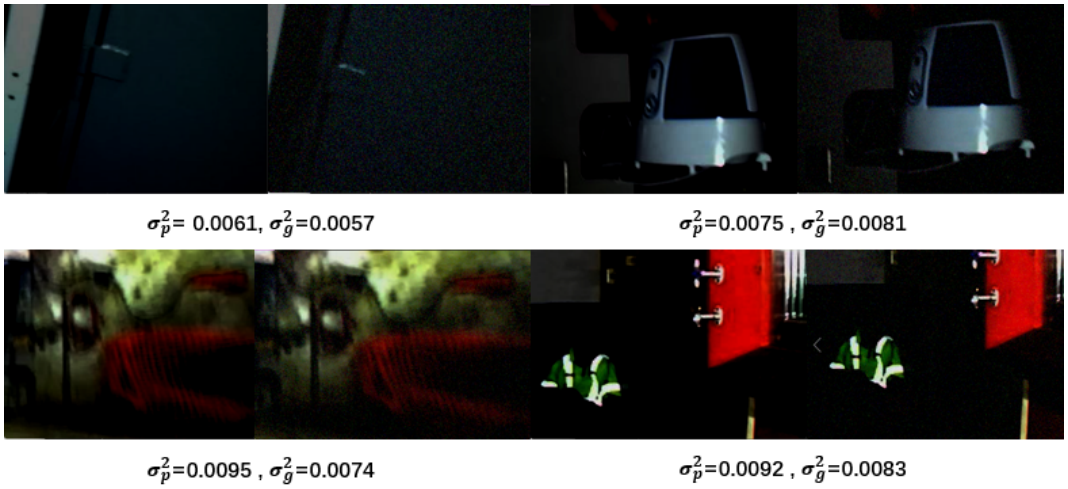


Figure 5.4: Several Pairs of Darken Image(Left) and Darken Image with Noise(Right), images are zoomed by 142%. Some other images with noise is provided in GitHub

5.1.2 Summary

In this subsection, this report presents two steps for creating a low-light database: Darken and adding noise. This report integrates these two steps through Matlab code. After that, 3359 images in the PST900 database were batch processed. The LLDSD of paired low-light noisy images is obtained.

This database will be used to train the following neural network:

- Normal Light Semantic Segmentation Network;
- Low Light Semantic Segmentation Network;
- Low Light Image Enhancement Network;
- Enhanced Image Semantic Segmentation Network.

5.2 Low Light Images Enhancement

5.2.1 Model parameters and results

The dataset used in this report to train the DLSR network is the previously proposed LLDSD, which contains 3359×3 underexposed images, and 3359 corresponding images of the PST900 dataset are used as Ground Truth.

The training mode of this report is as follows. 2015 images (60%) are randomly selected from PST900 as the ground truth of the training set. The low-light images (2015×3)

corresponding to these selected ground truth images are used as input images for training. In order to avoid the overfitting problem, the training samples are augmented using the method mentioned in the previous chapter.

The method used in this article is implemented on the PyTorch framework and uses the Adam optimizer[68]. Due to the limitation of the GPU computing power used, the total training epochs is 100, and the batch-size is set to 1. The model starts training with a learning rate of 0.0003, using CosineAnnealingLR for learning rate decay. The GPU model used is GTX 1060Ti, the memory is 4G, and the total training time is 168 hours.

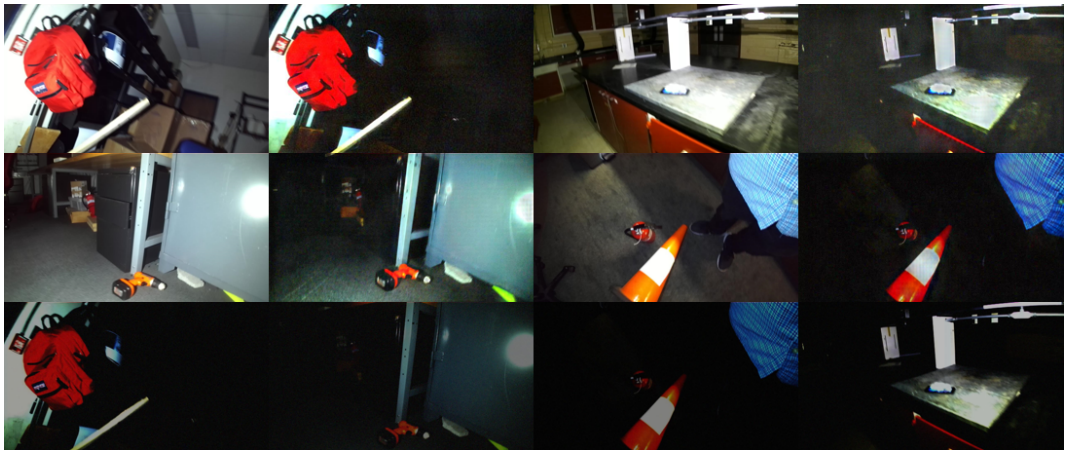


Figure 5.5: Some examples of the enhanced dataset compared to original images in PST900[3]. In each group of pictures, the original image is on the left, and the picture on the right is enhanced by the trained DSLR network on low light images. The last row is the same low light images in LLDSD

Finally, through the trained DLSR network, we perform low-light enhancement on the LLDSD dataset proposed in this report, and obtain a new dataset: Enhanced-LLDSD. Some examples of the enhanced dataset is shown in Figure 5.5. In the next section, this dataset and the PST900, LLDSD dataset will be used to train the semantic segmentation network and the results will be compared.

5.2.2 The performance of DSLR on LLDSD

To demonstrate the effectiveness of the trained image enhancement network, we compare the trained network with five traditional and deep learning based methods, namely LIME[69], MBLLEN[23], Zero-DCE[70], Retinex-Net[37] and LDR[71](among them MBLLEN, Zero-DCE, Retinex-Net and DSLR are learning based methods, the other two methods are traditional mehods). The above methods are all open source on Github, but due to time constraints, except for the DLSR network, the weights used by other deep learning-based methods are the original weights provided by the paper. All experiments testing each method are performed on a laptop with an Intel i7-6700K 4.0GHz CPU and NVIDIA Ge-

Force 1060Ti GPU.

Method	LLDSD	LIME	LDR	Retinex-net	ZERO-DCE	MBLLEN	DSLR
PSNR	7.892	14.127	11.014	12.488	15.175	18.032	22.678
SSIM	0.358	0.383	0.18	0.236	0.559	0.683	0.853

Table 5.1: Comparison on low-light images enhancement.In the process of judging the performance of DSLR, we mainly refer to two parameters, PSNR and SSIM.

In Table 5.1, we use two more commonly used image evaluation parameters (PSNR, SSIM) to judge the effect of different image enhancement methods on LLDSD[36]. PSNR is the ratio of the maximum power of the signal to the noise power, which measures the quality of the reconstructed image. The higher the PSNR index, the better the image quality. SSIM is an indicator to measure the similarity of two images, and its value range is [0, 1]. The larger the value of SSIM, the smaller the degree of image distortion and the better the image quality.

We calculate PSNR and SSIM by comparing the enhanced results of each image in the test set and the original images in the PST900 dataset, and finally average all the results to obtain the final result. From the results, we can see that, among the centralized methods compared in this report, both parameters of the method we use show that DSLR is a better image enhancement method on the LLDSD dataset.

An example of enhancement results on the LLDSD dataset is shown in Figure 5.6. From the pictures we can see that the two traditional methods, LDR and LIME, although they improve the visibility of low-light images to some extent, do not restore the relevant colors.

By looking at the results of the four deep learning-based methods, they can generally restore the brightness and color of a wide range of images. However, the colors in the enhanced results of Retinex-Net and Zero-DCE are noticeably faded after recovery and appear to be slightly overexposed. In addition, the noise of the picture seems to have increased.

The performance of MBLLEN is relatively good, but the image is still underexposed, but the noise of the image is not as much as the previous two methods. Compared with other methods, DSLR recovers the input image more successfully without generating excessive noise.

However, since other networks have not been trained on the LLDSD dataset proposed in this report, the better performance of DLSR may have certain limitations. But all in all, the trained DLSR can better complete the task of low light recovery.

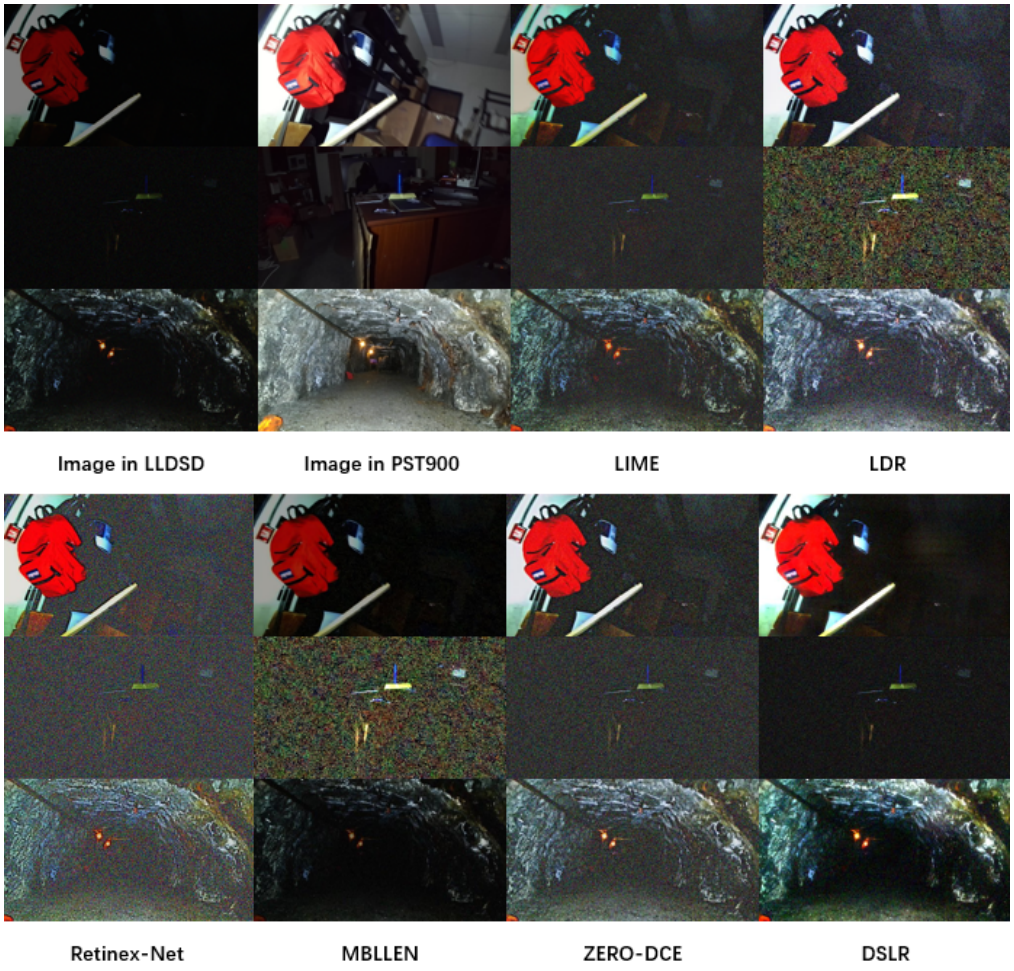


Figure 5.6: Comparison of Enhancement Result. The first two columns show the images in LLDSD and PST900, respectively. The other columns shows the enhanced result of these images.

5.3 Semantic Segmentation Network

In order to complete the task of semantic segmentation of low-light images, this report designs and completes a deep learning network based on PSPNet. In order to verify the effect of low-light image enhancement and semantic segmentation network, this chapter designs the following two experiments.

- Semantic Segmentation with LLDSD
- Semantic Segmentation with enhanced-LLDSD

The first experiment is to verify that PSPNet can perform the task of semantic segmentation of low-light images.
The second experiment is to verify whether the enhanced low-light images have better performance in the semantic segmentation network.

The experiments in this section are completed under the computer platform of Windows operating system, Intel Core i7-6700K 4.00GHz CPU processor, 16GB memory, and NVIDIA GTX1060 graphics card with 4GB memory. Use the PyTorch deep learning framework under Python3 to train and test the network.

The datasets used in this chapter, including training set and validation set, are generated based on the LLDSD dataset proposed in this report using the method introduced in Chapter 5. The data set was divided using the following method: The images in LLDSD were randomly divided into three groups, of which 80% were training set, 10% were validation set and 10% were test set. In order to ensure the fairness of training, in the process of training PSPNet with three different datasets, the contents of training set, test set and validation set are all the same. That is to say, except for brightness and noise, there is no difference in the content of each training dataset. Each group of pictures contains 3 pictures, namely low-light disaster scene picture, enhanced low-light disaster scene picture and semantic segmentation labels. The labels for semantic segmentation are provided by the PST dataset.

5.3.1 Model parameters

The three experimental software and hardware environments designed in this chapter are exactly the same. The three experiments use the same training and validation parameters to ensure fair results. The parameter settings are shown in Table 5.2.

Parameters	Setting
Backbone	Mobilenet
Initial learning rate	0.01
Weight decay	cos
Optimizer	SGD
Momentum	0.9
Downsample Factor	16
Freeze Epoch	50
Freeze batch size	4
UnFreeze Epoch	50
Unfreeze batch size	2
Training set	3023
Validation set	335
Test set	335

Table 5.2: Parameter Setting of PSPNet in this report.

The model starts training with a learning rate of 0.001, using CosineAnnealingLR for

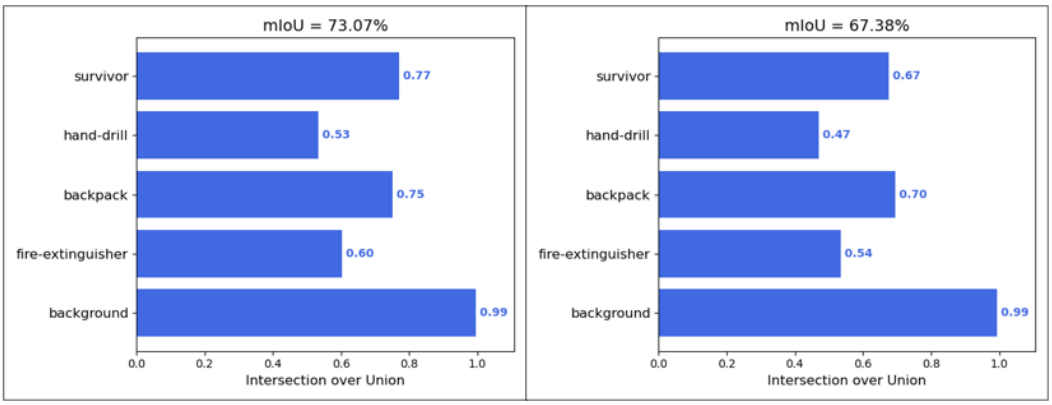
learning rate decay. The optimizer uses the SGD optimizer, total epochs are set to 100 times, and the Batch size is 2. The epochs of the model freezing training are set to 50, and the Batch size is 4. After the training is completed, all parameters are saved as .pth files.

In the experiments of this report, the parameters of the network and the division of the dataset are the same to eliminate the influence of the parameters and datasets on the experimental results. During training and testing, the mean intersection over union (MIoU), which determines the effect of segmentation by calculating the intersection of ground truth and predicted segmentation, is used to evaluate the semantic segmentation results.

5.3.2 Semantic Segmentation with PST900

In order to judge whether the low-light enhancement and semantic segmentation methods used in this report can meet the requirements in the semantic segmentation task of low-light images of disaster scenes, this report firstly trains the semantic segmentation network on the PST900 dataset. The training results can be used to judge the difference between the performance of the same semantic segmentation network in normal light conditions (PST900) and low light conditions (LLDSD).

In addition, as mentioned in the previous section, this report has made some changes in the loss function of the original PSPNet, so this report also trains the original PSPNet with the PST900 dataset to observe whether the changes in this report are effective.



(a) mIoU of changed PSPNet on PST900 (b) mIoU of original PSPNet on PST900

Figure 5.7: mIoU when training with PST900 on original PSPNet and changed PSPNet. Within the training, all the parameters are set to be the same to both networks.

As shown in Figure 5.7, the PSPNet in this report is trained on the PST900 dataset with an mIoU of 73.07% and mIoU on original PSPNet is 67.38%. At the same time, according to the figure, it is not difficult to find that the modified PSPNet performs better than the original PSPNet on all labels. Therefore, the performance of the changes to PSPNet in this report on the PST900 dataset can be called modified PSPNet.

Among the four labels predicted in this report, survivor and backpack have better performance. This may be due to the large area of survivors and backpacks in the pictures of the PST database. It may also be because the survivors in the database pictures are all wearing reflective clothes of the same color, and the backpacks are also of the same color (red). These reasons may lead to these two categories having easier-to-learn features and thus higher accuracy.

This conclusion can also be verified by the prediction result of the remaining two labels. Especially hand-drill, compared with the other three labels, hand-drill has the smallest volume, and has multiple colors (yellow and black) at the same time, and is more affected by lens shake, as shown in the Figure 5.8. Therefore, the final IoU is only 53%. Similar to this is the category of fire extinguishers, but due to its relatively large size, the accuracy is significantly improved compared to hand-drill, but the gap with the other two labels still exists. The reason for the improvement probably because the loss function we use in this report pays more attention to the samples that are difficult to learn.

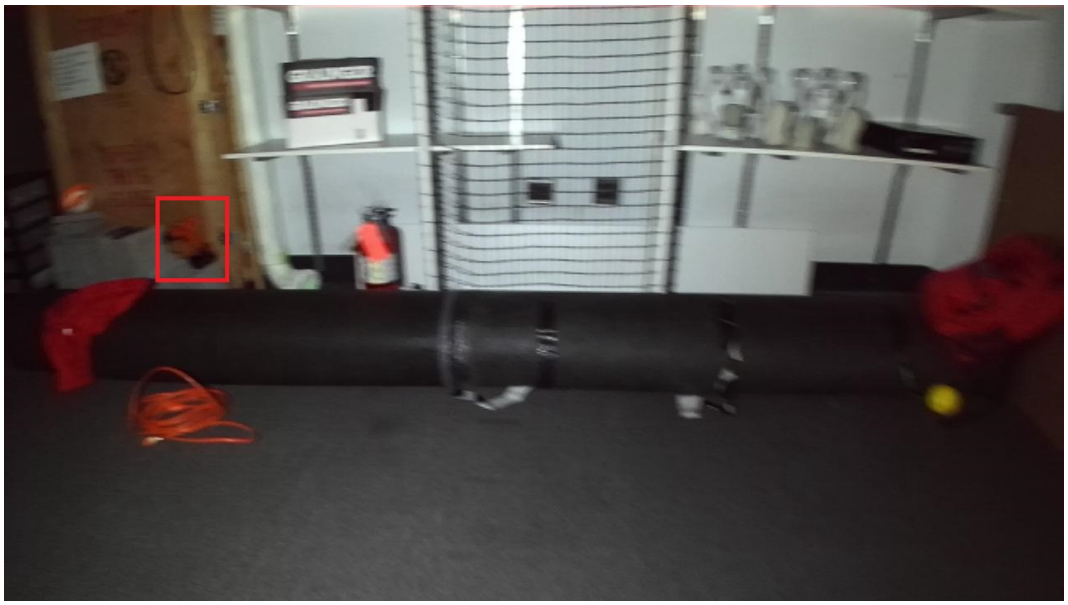


Figure 5.8: The blurred hand-drill(in the red box) in the PST900 dataset due to lens shake. Compared to the fire extinguishers in the middle the blurred hand-drill is hard to find under this condition.

In Figure 5.9 we show the semantic segmentation results of modified PSPNet and original PSPNet. From the figure, we can see that both PSPNets can effectively classify the pixels in the scene, but the segmentation results lose some details, such as the top of the fire extinguisher and the middle part of the hand drill are not restored. Comparing the results of the two PSPNets, we can find that the modified PSPNet is better than the original PSPNet in terms of classification accuracy. For example, in the segmentation of the middle image, the original PSPNet incorrectly divides the fire extinguisher into backpacks.The reason why the original PSPNet divides the fire extinguisher into the backpack incorrectly may be because the fire extinguisher and the backpack have the same color. During the training process, the network preferentially learns larger samples.

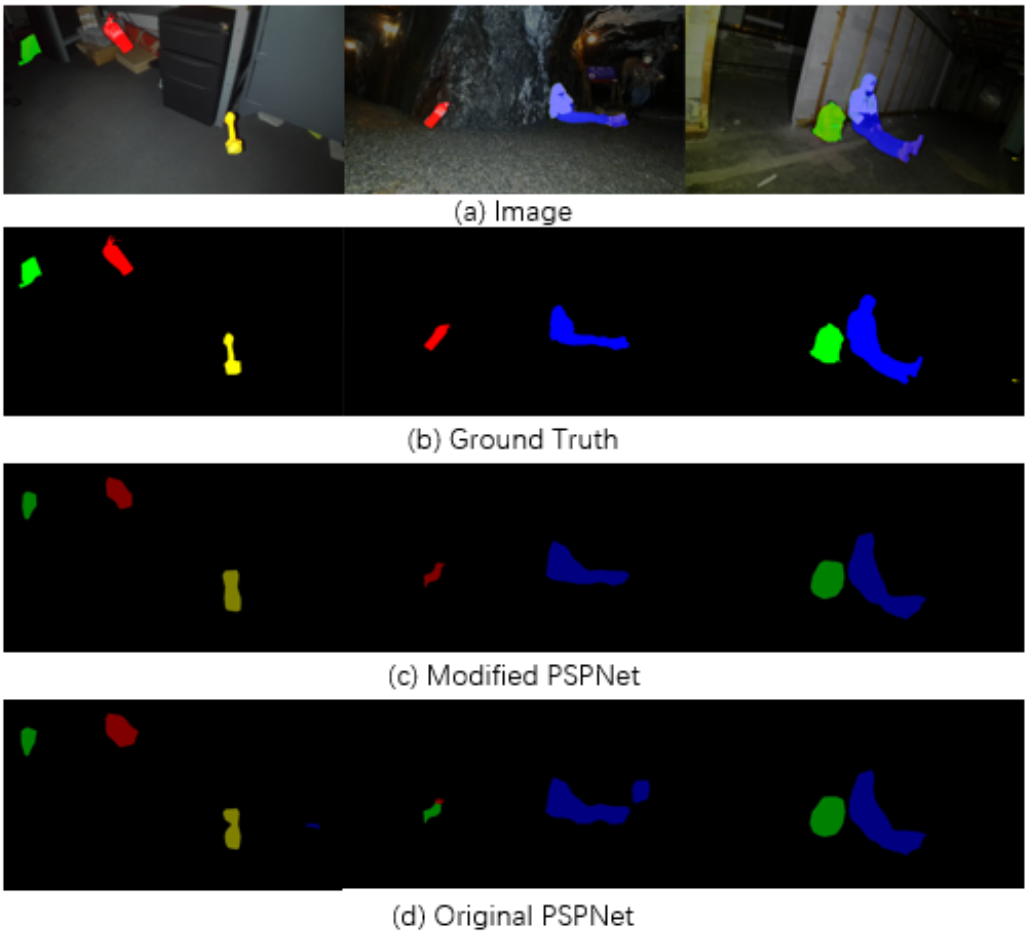


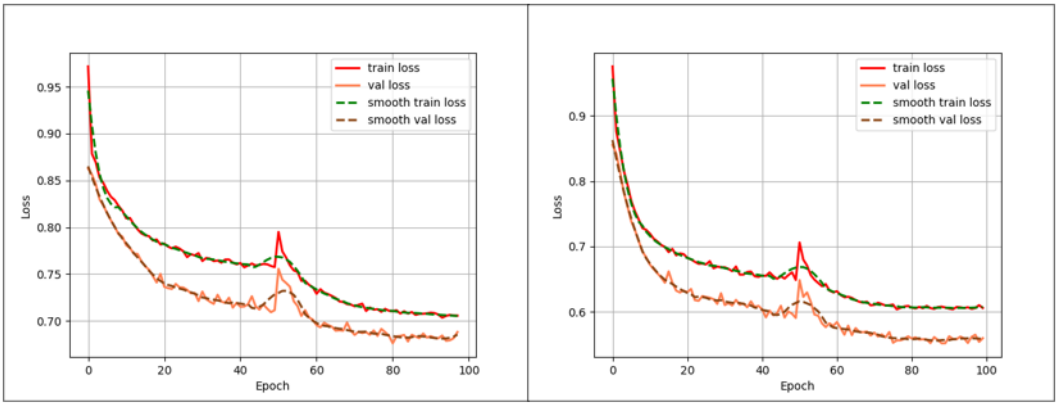
Figure 5.9: The first row of the image shows the segmentation result of the original image and Ground Truth, and the second row shows the Ground Truth with the background removed. The third and fourth rows are the semantic segmentation results of modified PSPNet and original PSPNet, respectively.

At the same time, because larger samples and smaller samples have some of the same characteristics, resulting in wrong classification. However, since the modified PSPNet replaces the loss function, the network pays more attention to the learning of small samples, so the phenomenon of misclassification will be effectively reduced.

In addition, in the classification results of survivors, original PSPNet mistook part of the background as survivors, and in terms of refinement, the effect of original PSPNet segmentation was not as good as modified PSPNet.

5.3.3 Semantic Segmentation with LLDSD

Figure 5.10 shows the semantic segmentation results, where red and orange represent the performance on the training and validation sets, respectively. It can be seen that at the 50th epoch, there is a large fluctuation in Loss. This is because the network is converted from frozen training to Unfreeze training, and the feature extraction network has changed. Overall, there was no over-fitting in the 100 epochs of training. At the same time, the Loss of the training set and the validation set are also reduced at the same time, which proves that the effect of the model gradually improves as the training progresses.



(a) Loss when training PSPNet with LLDSD (b) Loss when training PSPNet with PST900

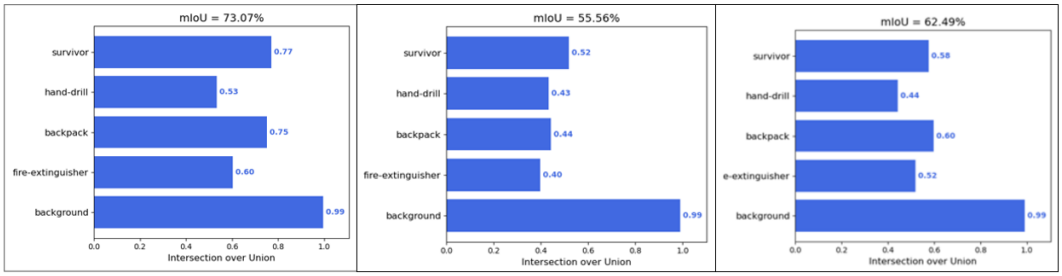
Figure 5.10: Loss when training net with LLDSD(Left) and PST900(Right). The train loss is higher than validation loss may because we did data augmentation when training.

Figure 5.11 shows the mIoU and mPA of semantic segmentation. From the figure, we can see that for the five categories marked in the PST database, the average IoU reached 55.56%, and the IoU of each category was also higher than 40%.

Compared with the results after training on the normal light image database, the performance of PSPNet on the LLDSD dataset has an overall decline, and only the mIoU of the survivor label has reached 50%. This may be because the low-light scenes and noise in the image of LLDSD affect the learning of the features by the neural network.

In addition, the mIoU of the two labels, survivor and backpack, has dropped very seriously. In comparison, the remaining two labels have dropped, but the drop is within 10%. For the label survivor, it may be that some features in the image (trouser color, facial details) are severely weakened in low light conditions, and in many cases only reflective clothes are clearly visible, as shown in the Figure 5.12.

For the label of backpack, as shown in Figure 5.13, the most obvious feature (color) is also



(a) mIoU of modified PSPNet on PST900 (b) mIoU of modified PSPNet on LLDSD (c) mIoU of modified PSPNet on Enhanced-LLDSD

Figure 5.11: mIoU of modified PSPNet trained on PST900(Left), LLDSD(Middle) and Enhancend LLDSD(Right). The training set, validation set and test set when training are the same.

Figure 5.12: Comparison of label "Survivor" between PST900 and LLDSD. The left side is the original image in PST900, the other one is image in LLDSD. In the images, we can see due to dark and noise some features of survivor are fused with the background.

significantly reduced due to the decrease in brightness, resulting in a significant decrease in mIoU. This also verifies the conjecture proposed in the previous subsection: the reason why the two labels of survivor and backpack are recognized under normal light conditions is higher because they have more obvious features.

According to Figure 5.14 we can see the result of training PSPNet directly on LLDSD. More misclassifications appeared in the segmentation results, such as identifying the survivor's legs as a backpack. And more unrecognized labels, such as the backpack is not recognized in the left picture, and the hand drill is recognized as two parts. This also happens in the middle picture, identifying the survivor's legs as the background.

The reason for these situations may be because, in the low-light database, more details are hidden. Especially when there are shadows in the normal light picture, in the corresponding low light picture, the visibility of the shadow part is lower. In addition, dark objects are difficult to distinguish from the background due to the low contrast of low-light images, for example, the black pants of volunteers were misidentified in the segmentation results.

Figure 5.13: Comparison of label "Backpack" between PST900 and LLDSD.The left side is the original image in PST900, the other one is image in LLDSD. Different from the former image, the color of backpack in LLDSD can still be distinguished with the background but the edge is hard to distinguish.

5.3.4 Semantic Segmentation with enhanced-LLDSD

Finally, we trained PSPNet on the augmented LLDSD dataset, and the training results are shown in Figure 5.11. In the figure, we can see that the accuracy of all four labels has improved compared to training on LLDSD before enhancement. Among them, the two labels of backpack and fire extinguisher have more than 10% accuracy improvement. Survivor has a 6% increase, but the hand-drill increase is only 1%.

Although there is a good improvement compared to the results on the low-light dataset, there is still a certain gap compared to the results on the PST900 dataset. In particular, there is still a 19% gap in the accuracy of the label survivor. The accuracy gap of the other three labels is about 10%-15%.

From the Figure 5.15 we can see the result of training PSPNet on Enhanced-LLDSD. Compared to PSPNet trained on LLDSD, the occurrence of misclassification is significantly reduced. And since the enhanced image restores more details, objects in shadow can also be effectively segmented. Only in some pictures, dark objects are confused with the background, but the degree and number are significantly reduced.

It can be seen that the method of enhancing low-light images first and then performing semantic segmentation can indeed improve the accuracy of semantic segmentation of low-light scenes. This confirms the result in paper[10], they also verified combine enhancement and segmentation can improve the performance of low light segmentation.

The failure cases are shown in Chapter 9, there are mainly 3 kinds of failure cases.

The first kind of failure is both modified PSPNet trained on PST900 and Enhanced LLDSD can not give correct perception. As shown in Figure 9.2, we can see that there is no labeled object in the image but the two networks both made some misclassification, the PSPNet trained on PST900 to predict the red object in the middle to hand-drill, and PSPNet trained on LLDSD predicts the highlighted area(right bottom part) as a survivor. This is probably because the brightness of the original image is low so both networks can’t get the correct result.

The second kind of failure is the PSPNet trained on PST900 can correctly predict but PSPNet trained on LLDSD get the wrong result. As shown in Figure 9.3, PSPNet trained on PST900 can predict the fire extinguisher in the middle of the image although still some pixels are not correctly predicted. However, as we can see in the sub-image the PSPNet trained on LLDSD didn't recognize the fire extinguisher at all. This is probably because the result of enhancement is not ideal, the enhancement result is totally distorted.

The third case is that PSPNet trained on PST900 can predict the right objects but at the same time there are some misclassifications. PSPNet trained on LLDSD can not predict the right object and also has some misclassifications. Like in Figure **??**, PSPNet trained on PST900 can correctly predict the backpack but mistakenly predict the light in the right bottom corner as hand-drill. PSPNet trained on LLDSD didn't recognise the backpack and predicted the object in the middle as a survivor. This may be because the enhancement network restores the middle of the image to a bright green color, which is the same color as the survivor's clothes, so it is predicted to be a survivor. In general, most of the errors in PSPNet trained on PST900 are due to the low brightness of the pictures in the PST900 dataset, and the network cannot judge the objects in the picture based on the learned features.

In addition to the above problems, most of the errors in PSPNet trained on PST900 are due to the unsatisfactory results of the enhancement network. The enhanced image still has color distortion, or excessively restores the brighter part of the picture, and the darker parts are not recovered enough, resulting in more shadows in the dark areas.. Even for some of the dark images in the PST900 dataset, the restored results are completely distorted, indicating the insufficient robustness of the augmentation network.

5.4 The performance of the combination of DSLR and PSPNet

In order to compare the effect of the combination of DLSR and PSPNet on low-light scene semantic segmentation, this section mainly compares the results of two experiments: PSPNet trained on the LLDSD dataset and PSPNet trained on the LLDSD dataset enhanced by the DLSR network. The quantified results are shown in the Table 5.3.
The reason why the two labels of Backpack and Fire Extinguisher can be significantly

Dataset:PST900/LLDSD					
Network	Fire-Extinguisher	Backpack	Hand-Drill	Survivor	**mIoU**
PSPNet(PST900)	0.6	0.75	0.53	0.77	0.7307
PSPNet(LLDSD)	0.4	0.44	0.43	0.52	0.5556
DLSR+PSPNet	0.52	0.6	0.44	0.58	0.6249

Table 5.3: The table shows the mIoU on each label and mean mIoU of each methods tested. PSPNet trained on PST900[3] dataset has the best performance within 3 methods. The third row is the mIoU of PSPNet trained on DSLR-enhanced low-light images, which shows a significant improvement compared to PSPNet trained on LLDSD (low-light unenhanced images).

improved may be because, with the enhancement of the images, more features of the

two labels are recovered, so that the feature extraction network can extract more features, which makes the network more accurate. The two labels and background are better differentiated. As shown in Figure 5.16, the backpack is difficult to see compared to the LLDSD dataset, and the enhanced image can clearly distinguish the backpack from the background.

The accuracy improvement of the Survivor label also benefits from this. However, as shown in Figure 5.16, although the upper body of the survivor in the enhanced picture has a higher degree of distinction from the background, the lower body is still not easily distinguishable from the background even after the enhancement. Therefore, although the enhancement can improve the accuracy to a certain extent, the improvement of the accuracy is not as much as that of the two labels of backpack and fire extinguisher.

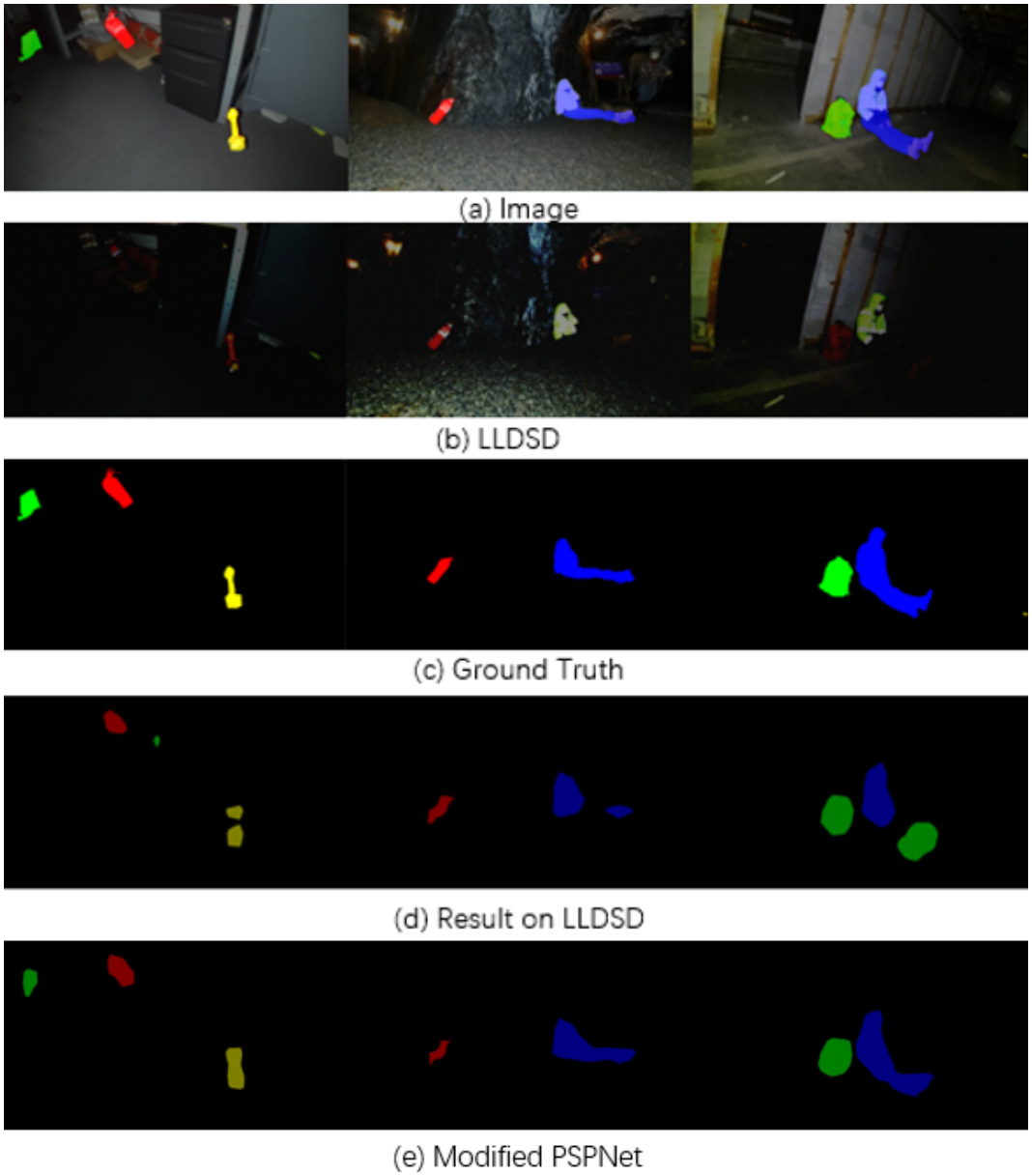


Figure 5.14: The first row of the image shows the segmentation results of the original image and Ground Truth, and the second row shows the same image in LLDSD. The third and fourth rows are the semantic segmentation results of Ground Truth and PSPNet in LLDSD, respectively.The last rowe is the result on PST900[3] by modified PSPNet

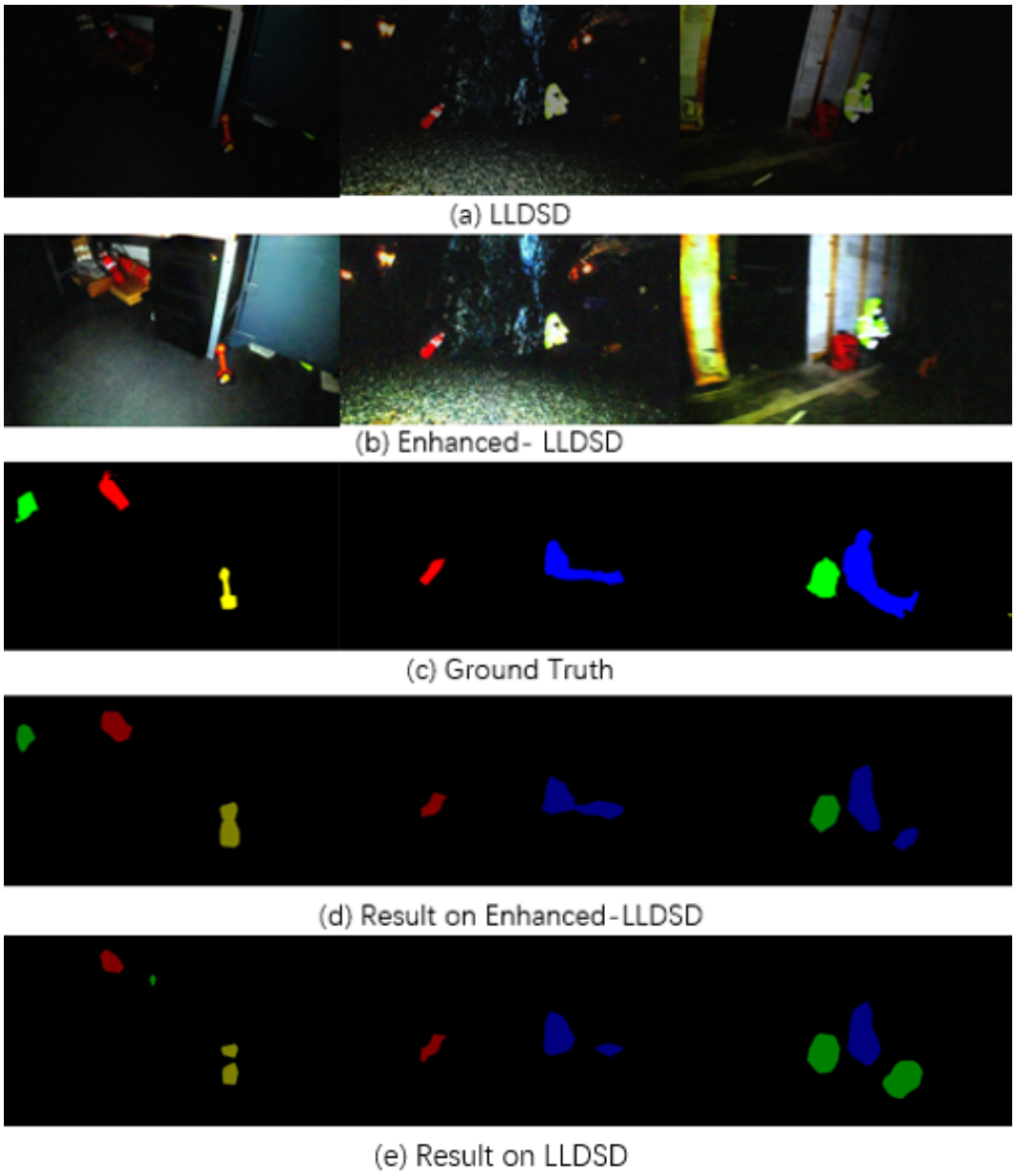


Figure 5.15: The first and second rows of the images show the same images from the LLDSD and Enhanced-LLDSD datasets, respectively. The third to fifth rows are the semantic segmentation results of Ground Truth, Enhanced-LLDSD and LLDSD, respectively.

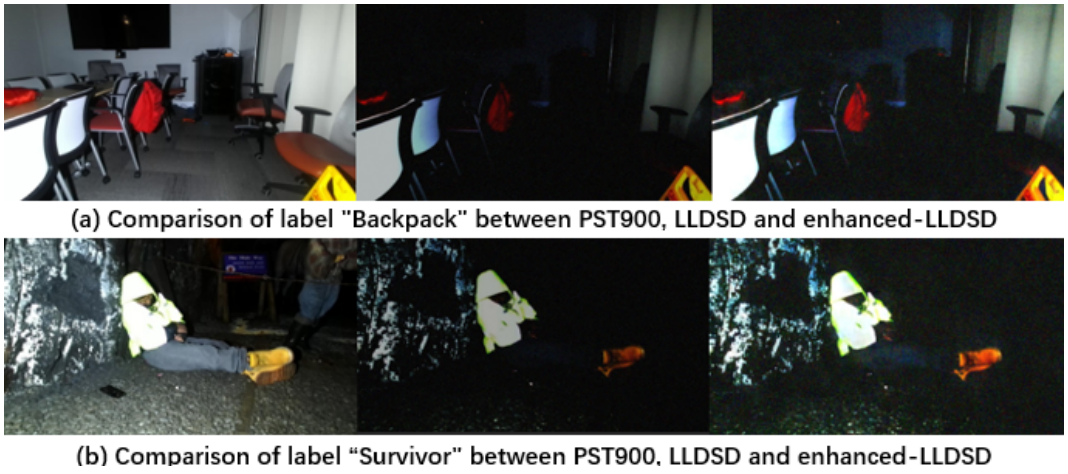


Figure 5.16: Comparison of label "Backpack"and "Survivor" between PST900, LLDSD and enhanced-LLDSD. The first column refers to the images containing label 'survivor' and 'backpack' in PST900[3] dataset, the second column shows the same image after low light synthetic in LLDSD. The last column shows the enhanced result of DSLR which is trained on LLDSD dataset.

6 - Discussion

We have shown that by combining PSPNet and DSLR, the accuracy of RGB image semantic segmentation results in low-light environments can be effectively improved. First, by gamma transforming and adding noise, we propose a low-light database-LLDSD, which can be used for semantic segmentation of disaster scenes, based on the PST900 database. We recover more details from low-light images by training a DSLR image enhancement network. The modified PSPNet is then trained on the augmented images to achieve semantic segmentation, and the accuracy is significantly improved compared to the direct semantic segmentation of low-light images.

This chapter will discuss different aspects of the system and will be divided into the following parts.

- The performance of DSLR on LLDSD;
- The performance of the modified PSPNet on PST900;
- The performance of the combination of DSLR and PSPNet on the task of low-light image semantic segmentation.

6.1 The performance of DSLR on LLDSD

When the trained DSLR network is used to enhance the LLDSD database, we can find from the quantitative and qualitative results that the effect of the DSLR network is slightly better than other comparative methods. But as mentioned in the previous chapter, this is likely because other network structures have not been trained on LLDSD.

When comparing the restored image with the original image, we can find that although some details have been restored better (color, light source, etc.), there is still a large gap between the brightness and the restoration of some small objects. Especially when there are shadow parts (caves, etc.) in the image, the enhanced image has a larger shadow part than the original image, and many details in the shadow part are hardly recovered. In addition, because the clothes of the volunteers in LLDSD can reflect light, and the lower body is dark pants, in the case of LED light source, the upper body brightness of the volunteers is higher than that of the lower body, so it can often be found that the enhanced volunteers have a higher brightness. The lower body is still missing some details, and it is even indistinguishable from the environment.

6.2 The performance of the modified PSPNet on PST900

Compared with original PSPNet, we make our modified PSPNet structure outperform original PSPNet on PST900 dataset by changing the loss function of the network. The mIoU of the modified PSPNet is 5% higher than the original PSPNet performance, and the most accurate improvement is the prediction of the label Suvivor, which is improved by 10%.

According to the comparison of modified PSPNet and original PSPNet in the previous chapter, we can see that modified PSPNet performs better than original PSPNet on the PST900 dataset after replacing the loss function. This is because regardless of the size of the image, the dice loss calculated in the region of the fixed-size positive sample is the same, and the supervision contribution to the network does not change with the size of the image, and because focal loss can focus more on harder-to-learn samples, so that in terms of miou and classification accuracy on small-size and infrequent labels which occupy fewer pixels, the modified PSPNet has a better improvement than the original PSPNet.

6.3 The performance of the combination of DSLR and PSPNet on the task of low-light image semantic segmentation

We find that the semantic segmentation task of low-light scenes can be partly solved by combining two methods, DSLR and PSPNet. The reason may be that for many low-light images, a large part of the details cannot be learned by the network due to problems such as brightness. When DSLR is used to enhance it, the image recovers more details, which help the neural network to perform feature learning and achieve the effect of improving accuracy.

Therefore, by combining DSLR and PSPNet, this report provides a solution capable of semantic segmentation in low light scene, while improving the accuracy of semantic segmentation in low light scene. This solves the first and third problems mentioned in Chapter3.

In addition, we also found an interesting phenomenon that the enhanced network has a higher accuracy improvement in recognizing medium-sized (backpack) categories. This may be because, in low-light conditions, smaller objects (hand-drill) are difficult to become more distinguishable by low-light enhancement, but medium-sized objects are more distinct from their surroundings due to enhancement, being able to recover more details also has the advantage of being easier to learn relative to small objects.

7 - Future Works

As mentioned, several aspects of the prototype could be improved. The first is the construction of the LLDSD database. Since there are few databases in disaster scenarios, and the labels recorded in these databases are not the same. For example, the PST900 dataset records four labels (fire extinguisher, hand-drill, survivor and backpack), and the UMA-SAR[13] dataset mainly records persons, vehicles, debris, and SAR activity on unstructured terrain. Simply fusing multiple SAR databases is not conducive to the training of semantic segmentation networks.

Therefore, only one of them (PST900) is used in this report when constructing the LLDSD database. This results in a smaller amount of data in the LLDSD database. Especially compared to more mature semantic segmentation databases such as ADE20[72] (training set: 20,210 images, validation set: 2,000 images, 150 labels), our database has considerable limitations in size and types of labels.

Furthermore, when building the database, we obtain low-light images by performing low-light synthesis on real images. But even with different methods (such as adding different forms of noise) to increase the realism of the synthesized image, there is still some gap between the synthesized image and the real low-light image. These gaps will lead to sub-optimal performance of the image enhancement methods and semantic segmentation networks trained in this report on real low-light scene semantic segmentation.

Therefore, the follow-up work on the LLDSD dataset may be to add more high-quality pixel-level annotations images or weakly annotated images. In addition, the reliability of LLDSD can be improved by collecting more images of low-light disaster scenes of real scenes. Considering the difficulty of collecting real low-light disaster scene images, more advanced methods such as CycleISP[73] can also be used to simulate low-light images.

The main problem with DSLR networks is that the performance and robustness are not good enough. It is found that its robustness is not enough, the lighting enhancement results are still different from the original image, and the ability to restore details in dark places is poor. Besides, the images with low brightness in PST900 have even lower brightness after synthesis. The enhancement results of these images often cause distortion of the whole image, so that the semantic segmentation network cannot recognize the target at all. The reason for this problem may be that the dataset used for training is synthesized and the dataset is not big enough. Therefore, future work may be to collect more paired images in disaster scene for training enhancement network.

For the semantic segmentation network, there are mainly the following ways to improve the performance. Due to the lack of computer performance we used during training, a smaller batch size and a lightweight backbone were used. Using a larger batch size may

make it easier for the network to converge and improve training performance. Replacing other backbones may also improve network performance. In addition, since this report replaces the model of the loss function to improve the accuracy, it may also be possible to improve the performance of the network by using one or several other loss functions, such as combining the loss function used in this report and the cross-entropy loss function.

8 - Conclusion

In this project we set out to create a low light disaster scene semantic segmentation method and answer the following questions from the problem formulation:

- How to use a set of algorithms to make rescue robots have the ability of semantic segmentation in dark disaster environment?
  - How can the low light disaster scene database be created?
  - How to do semantic segmentation in a dark environment?
  - Is it possible to improve the accuracy of semantic segmentation by doing some processing on dark pictures compared to directly train a semantic segmentation network on the normal light dataset or on the low light dataset?

In order to solve the problem that there is currently no database for low-light disaster scenes, based on the PST900[3] database, we performed low light synthesis on the database and built a new low-light disaster scene database-LLDSD.

Furthermore, we enhance the images in LLDSD through the DSLR network, and compare the results with some existing methods to verify the performance of the DSLR network on LLDSD, and generate a new dataset-Enhanced-LLDSD.

Furthermore, we modified the original PSPNet by using a combination of dice loss and focal loss rather than standard cross entropy loss, first comparing the performance of the network on PST900, the architecture achieves better performance among the compared methods on this dataset.

We then train a semantic segmentation network on LLDSD and Enhanced-LLDSD, respectively, and demonstrate that images enhanced by DLSR can significantly improve the performance of low-light image semantic segmentation tasks.

Bibliography

[1] Seokjae Lim and Wonjun Kim. Dslr: Deep stacked laplacian restorer for low-light image enhancement. *IEEE Transactions on Multimedia*, 23:4272–4284, 2021.

[2] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network, 2016.

[3] Shreyas S. Shivakumar, Neil Rodrigues, Alex Zhou, Ian D. Miller, Vijay Kumar, and Camillo J. Taylor. Pst900: Rgb-thermal calibration, dataset and segmentation network, 2019.

[4] Yosuke Yajima, Seongyong Kim, Jing Dao Chen, and Yong Cho. Fast online incremental segmentation of 3d point clouds from disaster sites. In *Proceedings of the 38th International Symposium on Automation and Robotics in Construction (ISARC)*. International Association for Automation and Robotics in Construction (IAARC), November 2021.

[5] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation, 2014.

[6] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation, 2015.

[7] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015.

[8] Steven W. Chen, Shreyas S. Shivakumar, Sandeep Dcunha, Jnaneshwar Das, Edidiong Okon, Chao Qu, Camillo J. Taylor, and Vijay Kumar. Counting apples and oranges with deep learning: A data-driven approach. *IEEE Robotics and Automation Letters*, 2(2):781–788, 2017.

[9] Shervin Minaee, Yuri Boykov, Fatih Porikli, Antonio Plaza, Nasser Kehtarnavaz, and Demetri Terzopoulos. Image segmentation using deep learning: A survey, 2020.

[10] Se Woon Cho, Na Rae Baek, Ja Hyung Koo, Muhammad Arsalan, and Kang Ryoung Park. Semantic segmentation with low light images by modified cyclegan-based image enhancement. *IEEE Access*, 8:93561–93585, 2020.

[11] Guiying Tang, Li Zhao, Runhua Jiang, and Xiaoqin Zhang. Single image dehazing via lightweight multi-scale networks. In *2019 IEEE International Conference on Big Data (Big Data)*, pages 5062–5069, 2019.

[12] Bolun Cai, Xiangmin Xu, Kui Jia, Chunmei Qing, and Dacheng Tao. DehazeNet: An end-to-end system for single image haze removal. *IEEE Transactions on Image Processing*, 25(11):5187–5198, nov 2016.

[13] Jesús Morales, Ricardo Vázquez-Martín, Anthony Mandow, David Morilla-Cabello, and Alfonso García-Cerezo. The UMA-SAR dataset: Multimodal data collection from a ground vehicle during outdoor disaster response training exercises. *The International Journal of Robotics Research*, 40(6-7):835–847, 2021.

[14] Hae-Gon Jeon, Sunghoon Im, Byeong-Uk Lee, Dong-Geol Choi, Martial Hebert, and In So Kweon. Disc: A large-scale virtual dataset for simulating disaster scenarios. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 187–194, 2019.

[15] Bibo Lu, Zebang Pang, Yanan Gu, and Yanmei Zheng. Channel splitting attention network for low-light image enhancement. *IET Image Processing*, 16(5):1403–1414.

[16] Haiyang Jiang and Yinqiang Zheng. Learning to see moving objects in the dark. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7323–7332, 2019.

[17] Chen Chen, Qifeng Chen, Minh Do, and Vladlen Koltun. Seeing motion in the dark. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3184–3193, 2019.

[18] Wenhan Yang Chen Wei, Wenjing Wang. Deep retinex decomposition for low-light enhancement. In *British Machine Vision Conference*, 2018.

[19] Jianrui Cai, Shuhang Gu, and Lei Zhang. Learning a deep single image contrast enhancer from multi-exposure images. *IEEE Transactions on Image Processing*, 27(4):2049–2062, 2018.

[20] Vladimir Bychkovsky, Sylvain Paris, Eric Chan, and Fredo Durand. Learning photographic global tonal adjustment with a database of input / output image pairs. In *CVPR 2011*, pages 97–104, 2011.

[21] Chen Chen, Qifeng Chen, Jia Xu, and Vladlen Koltun. Learning to see in the dark, 2018.

[22] Jiaying Liu, Dejia Xu, Wenhan Yang, Minhao Fan, and Haofeng Huang. Benchmarking low-light image enhancement and beyond. *Int. J. Comput. Vision*, 129(4):1153–1184, apr 2021.

[23] Jianhua Wu Feifan Lv, Feng Lu and Chongsoon Lim. Mbllen: Low-light image/video enhancement using cnns. *British Machine Vision Conference*, 2018.

[24] Feifan Lv, Yu Li, and Feng Lu. Attention guided low-light image enhancement with a large scale low-light simulation dataset, 2019.

[25] Kin Gwn Lore, Adedotun Akintayo, and Soumik Sarkar. Llnet: A deep autoencoder approach to natural low-light image enhancement, 2015.

[26] Ruixing Wang, Qing Zhang, Chi-Wing Fu, Xiaoyong Shen, Wei-Shi Zheng, and Jiaya Jia. Underexposed photo enhancement using deep illumination estimation. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6842–6850, 2019.

[27] Shangchen Zhou, Chongyi Li, and Chen Change Loy. Lednet: Joint low-light enhancement and deblurring in the dark, 2022.

[28] H.D. Cheng and X.J. Shi. A simple and effective histogram equalization approach to image enhancement. *Digital Signal Processing*, 14(2):158–170, 2004.

[29] Yeong-Taeg Kim. Contrast enhancement using brightness preserving bi-histogram equalization. *IEEE Transactions on Consumer Electronics*, 43(1):1–8, 1997.

[30] Yu Wang, Qian Chen, and Baeomin Zhang. Image enhancement based on equal area dualistic sub-image histogram equalization method. *IEEE Transactions on Consumer Electronics*, 45(1):68–75, 1999.

[31] Edwin H. Land and John J. McCann. Lightness and retinex theory. *J. Opt. Soc. Am.*, 61, Jan 1971.

[32] D.J. Jobson, Z. Rahman, and G.A. Woodell. A multiscale retinex for bridging the gap between color images and the human observation of scenes. *IEEE Transactions on Image Processing*, 6(7):965–976, 1997.

[33] Xueyang Fu, Delu Zeng, Yue Huang, Xiao-Ping Zhang, and Xinghao Ding. A weighted variational model for simultaneous reflectance and illumination estimation. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2782–2790, 2016.

[34] Xiaojie Guo, Yu Li, and Haibin Ling. Lime: Low-light image enhancement via illumination map estimation. *IEEE Transactions on Image Processing*, 26(2):982–993, 2017.

[35] Chongyi Li, Chunle Guo, Linghao Han, Jun Jiang, Ming-Ming Cheng, Jinwei Gu, and Chen Change Loy. Low-light image and video enhancement using deep learning: A survey. 2021.

[36] Shervin Minaee, Yuri Boykov, Fatih Porikli, Antonio Plaza, Nasser Kehtarnavaz, and Demetri Terzopoulos. Image segmentation using deep learning: A survey, 2020.

[37] Wenhan Yang Chen Wei, Wenjing Wang. Deep retinex decomposition for low-light enhancement. In *British Machine Vision Conference*. British Machine Vision Association, 2018.

[38] Yonghua Zhang, Jiawan Zhang, and Xiaojie Guo. Kindling the darkness: A practical low-light image enhancer. MM '19, 2019.

[39] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network, 2016.

[40] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[41] Liang-Chieh Chen, Yi Yang, Jiang Wang, Wei Xu, and Alan L. Yuille. Attention to scale: Scale-aware semantic image segmentation, 2015.

[42] Pauline Luc, Camille Couprie, Soumith Chintala, and Jakob Verbeek. Semantic segmentation using adversarial networks. 2016.

[43] Yuanbin Wang and Jieying Ren. Low-light forest flame image segmentation based on color features. *Journal of Physics: Conference Series*, 1069:012165, aug 2018.

[44] Vladimir Haltakov, Jakob Mayr, Christian Unger, and Slobodan Ilic. Semantic segmentation based traffic light detection at day and at night. In Juergen Gall, Peter Gehler, and Bastian Leibe, editors, *Pattern Recognition*. Springer International Publishing, 2015.

[45] Orcan Alpar. Corona segmentation for nighttime brake light detection. *IET Intelligent Transport Systems*, 10(2):97–105, 2016.

[46] Tarun Kancharla, Pallavi Kharade, Sanjyot Gindi, Krishnan Kutty, and Vinay G Vaidya. Edge based segmentation for pedestrian detection using nir camera. In *2011 International Conference on Image Information Processing*, pages 1–6. IEEE, 2011.

[47] Dengxin Dai and Luc Van Gool. Dark model adaptation: Semantic image segmentation from daytime to nighttime. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 3819–3824. IEEE, 2018.

[48] Lei Sun, Kaiwei Wang, Kailun Yang, and Kaite Xiang. See clearer at night: towards robust nighttime semantic segmentation through day-night image conversion. In *Artificial Intelligence and Machine Learning in Defense Applications*, volume 11169, page 111690A. International Society for Optics and Photonics, 2019.

[49] Abhinav Valada, Johan Vertens, Ankit Dhall, and Wolfram Burgard. Adapnet: Adaptive semantic segmentation in adverse environmental conditions. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4644–4651. IEEE, 2017.

[50] Se Woon Cho, Na Rae Baek, Ja Hyung Koo, Muhammad Arsalan, and Kang Ryoung Park. Semantic segmentation with low light images by modified cyclegan-based image enhancement. *IEEE Access*, 8:93561–93585, 2020.

[51] Terezinha Medeiros Gonçalves de Loureiro, Ketan Brodeur, Genevieve Schade, and Brito. Effect of the decrease in luminance noise range on color discrimination of dichromats and trichromats. *Frontiers in Behavioral Neuroscience*, 12, 2018.

[52] Alireza Zolghadr-E-Asli and Siamak Alipour. An effective method for still image compression/decompression for transmission on pstn lines based on modifications of huffman coding. *Computers & Electrical Engineering*, 30(2):129–145, 2004.

[53] R.C. Gonzalez and R.E. Woods. *Digital Image Processing*. Prentice Hall, 2002.

[54]

[55] Xiangcheng Chen. On schottky noise and shot noise, 2018.

[56] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. 2018.

[57] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. 2018.

[58] Zhilu Zhang and Mert R. Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels, 2018.

[59] Carole H. Sudre, Wenqi Li, Tom Vercauteren, Sebastien Ourselin, and M. Jorge Cardoso. Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations. In *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, pages 240–248. Springer International Publishing, 2017.

[60] Florian Chabot, Quoc-Cuong Pham, and Mohamed Chaouch. Lapnet : Automatic balanced loss and optimal assignment for real-time dense object detection, 2019.

[61] Shruti Jadon. A survey of loss functions for semantic segmentation. In *2020 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*. IEEE, oct 2020.

[62] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection, 2017.

[63] Maxim Tatarchenko, Alexey Dosovitskiy, and Thomas Brox. Multi-view 3d models from single images with a convolutional network, 2016.

[64] Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Semantic foggy scene understanding with synthetic data. *International Journal of Computer Vision*, 126(9):973–992, Sep 2018.

[65] Feifan Lv, Yu Li, and Feng Lu. Attention guided low-light image enhancement with a large scale low-light simulation dataset, 2019.

[66] Shi Guo, Zifei Yan, Kai Zhang, Wangmeng Zuo, and Lei Zhang. Toward convolutional blind denoising of real photographs. 2018.

[67] Shreyas S. Shivakumar, Neil Rodrigues, Alex Zhou, Ian D. Miller, Vijay Kumar, and Camillo J. Taylor. Pst900: Rgb-thermal calibration, dataset and segmentation network, 2019.

[68] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2014.

[69] Xiaojie Guo, Yu Li, and Haibin Ling. Lime: Low-light image enhancement via illumination map estimation. *IEEE Transactions on Image Processing*, 26(2):982–993, 2017.

[70] Chunle Guo Guo, Chongyi Li, Jichang Guo, Chen Change Loy, Junhui Hou, Sam Kwong, and Runmin Cong. Zero-reference deep curve estimation for low-light image enhancement. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pages 1780–1789, June 2020.

[71] Chulwoo Lee, Chul Lee, and Chang-Su Kim. Contrast enhancement based on layered difference representation of 2d histograms. *IEEE Transactions on Image Processing*, 22(12):5372–5384, 2013.

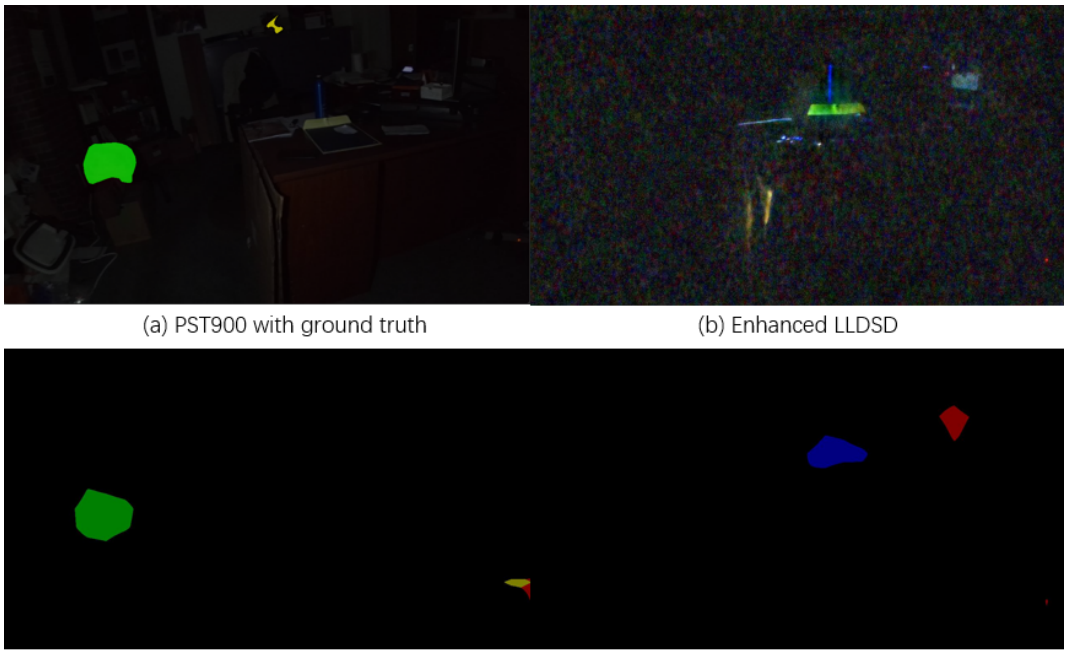
[72] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 633–641, 2017.

[73] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. Cycleisp: Real image restoration via improved data synthesis, 2020.

9 - Appendix

9.1 Failure case of modified PSPNet on Enhanced-LLDSD

For each figure, sub-figure a shows images in PST900 with ground truth. Sub-figure b shows images in Enhanced-LLDSD. Sub-figure c and d represent the segmentation result of modified PSPNet trained on PST900 and Enhanced-LLDSD, respectively.



(a) PST900 with ground truth

(b) Enhanced LLDSD

(c) Result on modified PSPNet(PST900)

(d) Result on modified PSPNet(Enhanced LLDSD)

Figure 9.1: Example of failure case 1. We can see that although the modified PSPNet trained on PST900 still misclassifies the object in the lower right corner, it can more accurately identify the backpack from the picture. However, the modified PSPNet trained on enhanced LLDSD not only fails to recognize the backpack, but also misclassifies the two brighter parts of the image.

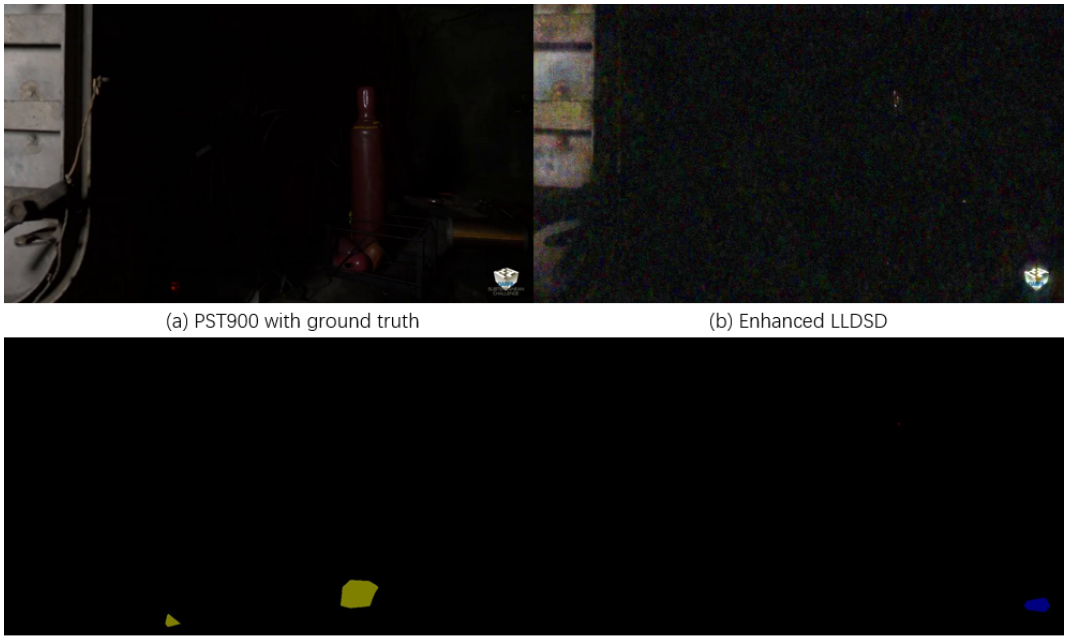


Figure 9.2: Example of failure case 2.According to the picture, we can see that there is no labeled item in the original picture of PST900, but the modified PSPNet still recognizes the red object as hand-drill. The modified PSPNet trained on enhanced LLDSD identifies the highlighted part in the lower right corner as survivor. The modified PSPNet trained on enhanced LLDSD may not misclassifies the objects in the middle due to the low brightness of the enhanced results.

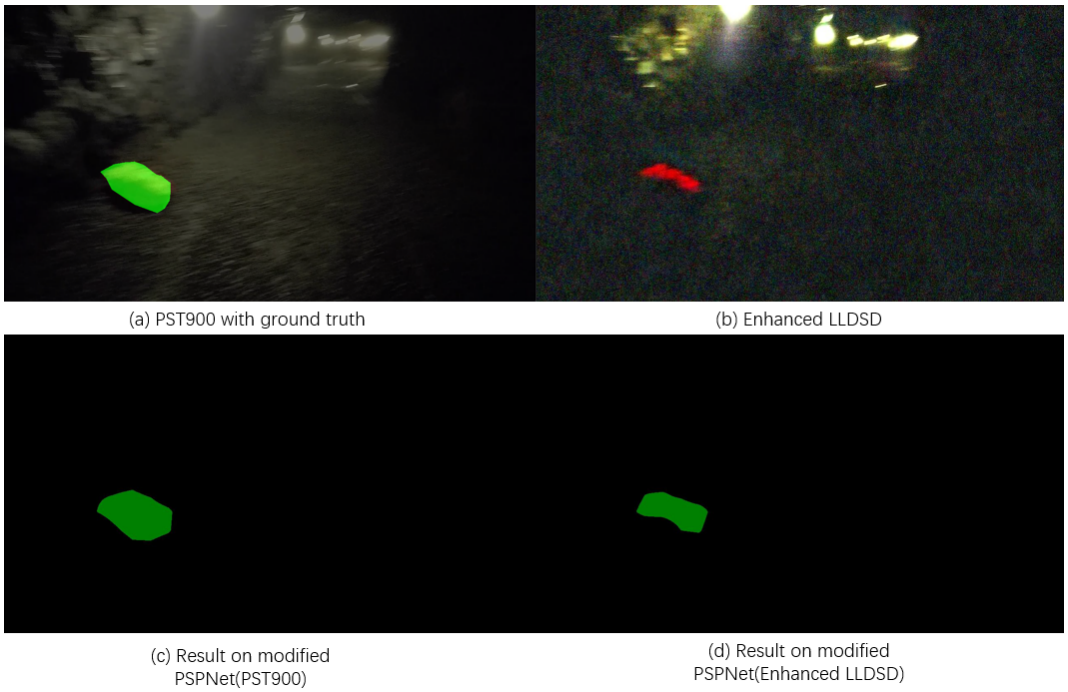


Figure 9.3: According to the picture, we can see that both PSPNets effectively identify the backpack. However, due to the shadows in the enhanced images, there is still a certain gap between the results of PSPNet recognition and ground truth.

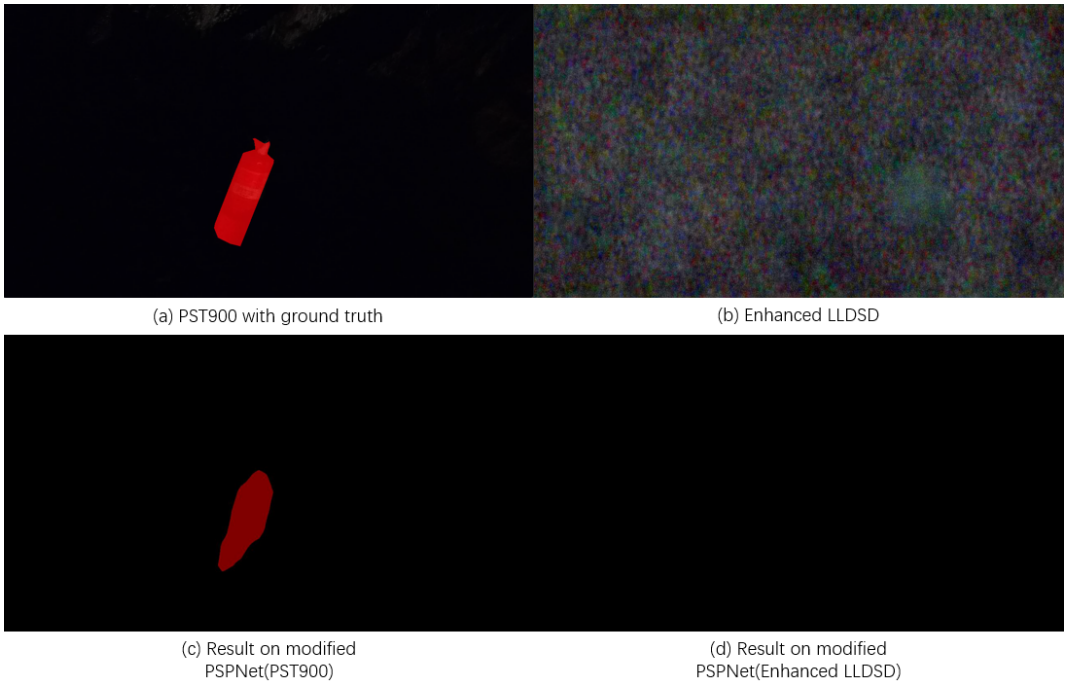


Figure 9.4: Example of failure case 3.From the pictures we can see that the original image in the PST900 is extremely low in brightness, making it difficult to distinguish the fire extinguisher from it. The modified PSPNet trained on PST900 is still able to correctly classify the fire extinguisher in the picture, but there are still some pixels that are not correctly identified. In this case, we can see that the enhanced image is severely distorted, and we can't even distinguish any details of the image. Therefore, the modified PSPNet trained on enhanced LLDSD cannot correctly identify this image.