Project Report

Single-frame Infrared Small Target (SIRST) Detection

Course: IT3320E - Introduction to Deep Learning

Authors: Group 16

Nguyen Tong Minh - 20204885
Hoang Tran Nhat Minh - 20204883
Nguyen Hoang Phuc - 20204923
Ho Minh Khoi - 20204917
Truong Quang Binh - 20200068

Advisor: Prof. Nguyen Hung Son

Academic semester: 20221

Abstract

Infrared small target detection has been a challenging task due to the weak radiation intensity of targets and the complexity of the background. Traditional methods using hand-designed features are usually effective for specific background but pose some problems in other complex infrared scenes. Our work proposes some deep learning based approaches on single-frame infrared small target (SIRST) detection in order to exploit the unexpected methods that potentially lead to more adaptive and accurate solutions. Distinct artificial neural networks are trained through thousands of infrared images in order to obtain the patterns of desired tiny, disrupted targets and then suppress the other non-target regions. Extensive experiments demonstrate that the proposed methods potentially handle effectively the variety and difficulty of this problem, compared to common fixed algorithms, in terms of visual and quantitative evaluation metrics.

We would like to express our sincere gratitude to Prof. Nguyen Hung Son for giving us an opportunity to work on this project of infrared small target detection. Throughout your detailed lectures and great advices on training artificial neural networks, we wanted to say thank you for your valuable words that lead and motivated us through the hardships. We are also extremely grateful to the teaching assistants who guided us through Pytorch framework and provided fruitful hands-on laboratory. Preparing this project in collaboration with you and your assistants was greatly a refreshing experience.

Contents

1 Introduction

Infrared small target detection is one of the research hot-spots in the field of military reconnaissance, which is widely used in the early-warning system, precision-guided weapon, surveillance and tracking system, and maritime surveillance system. Based on the infrared target monitoring system, it uses the infrared radiation differences between background and target to detect the target. Comparing with the radar detection system, the infrared target detection system has strong concealment, easy portability and can detect radar blind area. However, owing to atmospheric disturbance, optical scattering and diffraction, the target has low radiation intensity, low signal to-clutter ratio (SCR) and lack of shape and texture information. It thus easily drowns into the background [1]. Besides, complex natural scenes such as cloud edges and waves usually introduce false positive in detection; and furthermore in many cases, the assumptions of static backgrounds do not apply. Therefore, the single-frame infrared small target detection is a valuable task but comes with great challenges.

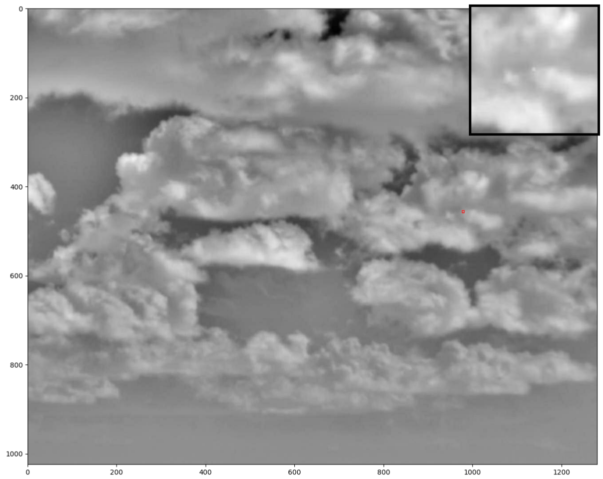

Figure 1: A single frame with the ground-truth box (zoomed in on the top left).

Across this work, our goals of infrared small target detection are to improve the detection rate (of true positive cases) and reduce the false alarm rate (of false positive cases). In general, infrared small target detection methods are mainly divided into two categories: one is track before detect (TBD) method based on sequence images, and the other is detect before track (DBT) method based on a single-frame image.

TBD methods detect the targets by processing several adjacent frames, which is suitable for the situation that the background changes slowly and the target track is continuous in the infrared sequence. However, multi-frame detection is hard to realize in hardware because of its computational complexity and memory consumption. On the contrary, the DBT method is real-time, low in complexity and easy to implement. It preprocesses the single-frame image, extracts suspicious targets by threshold segmentation, and confirms the targets according to the moving track of the targets on the image sequence. Over the past few decades, many methods based on single-frame detection are proposed.

Filter-based methods use different filter templates to suppress background, such as Max–median/Max–mean filters [2], morphology filtering [3], Top-Hat transform [4], two-dimensional least mean square (TDLMS) filter [5]. It is hard to obtain the universal template parameters and organize elements without prior information, meaning that adaptability and robustness are hard to be guaranteed.

Most of the traditional methods detect targets by hand-designed features, which are usually only effective for a certain background. However, the real infrared small target detection scenes include sky, cloud, sea and various building backgrounds. The traditional method using only artificial features obviously lacks robustness in the complex and changeable application scenes, while the method based on deep learning can extract features by convolutional neural network (CNN), which can make up for the lack of feature and the difficulty of feature description in infrared image.

Deep learning has made great progress in the field of target detection recently, there are mainly two types of detection pipelines: two-stage region-based method and one-stage method. Our work then try to approach the problem of infrared images with both two types of network structure.

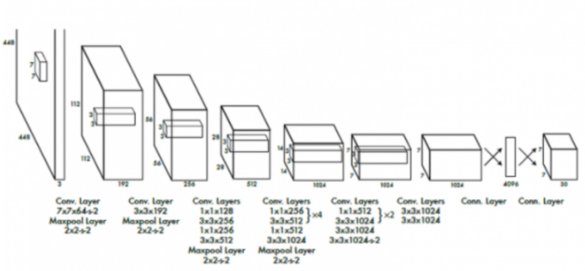

Figure 2: A one-stage architecture model (YOLO).

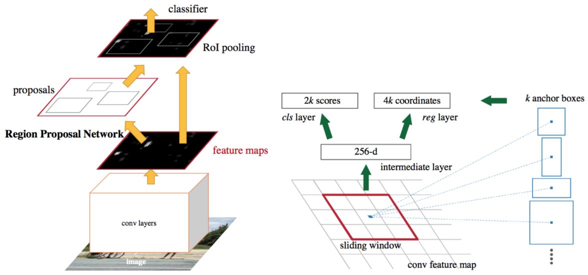

Figure 3: A two-stage architecture model (Faster RCNN).

This report contains 5 main sections. Section 1 is Introduction. Section 2 is Methodology - the most important part in this report, where we describe and explain the network architectures used to detect the infrared small targets. Section 3 is Experiment, where we tell further about the dataset chosen, training settings of each model and evaluation results. We then conclude the whole report in section 4.

2 Methodology

Our methods proposed come from basic convolutional neural networks to very complex architectures. For the first two networks, the former is an one-stage model, composed of common layers, that processes end-to-end for single frames. The latter is a two-stage network, resembling the structure of RCNN with two separated modules. An advance of its is that the regional proposal module is replaced with a corner detection algorithm instead of selective search. The latter ones, segementation models, are then proposed to not only preserve the end-to-end feature of the first network but also attain a better result, compared to the second network. Finally, we compare all of the proposed methods with a famous fully handcrafted method in this area, WLCV.

2.1 Weighted Local Coefficient of Variables

For small target detection, it was long-ago believed that deep learning models were inefficient since they require complex calculations and take a lot of time to process. In 2022, Junmin Rao et al. [6] suggested a DBT framework for SIRST dataset [**https://github.com/YimianDai/sirst**]. The pipeline of detection is proposed as in the following figure.

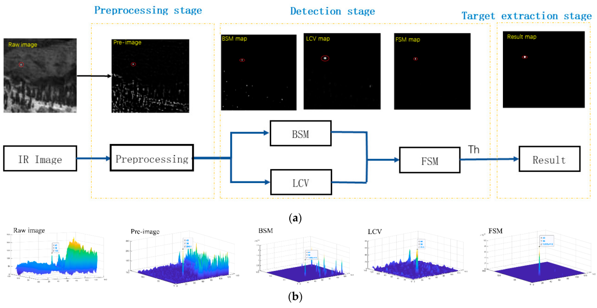

Figure 4: The detection pipeline.

This pipeline may work well with SIRST dataset, however, for our dataset, there should be several modifications. In the following sections, we will discuss what we do and what to modify.

2.1.1 Preprocessing

In the original paper, preprocessing consists of 2 steps: A Weiner filter followed by a High-pass Laplacian filter. The purpose of those filters was to spot areas with high contrast, which gives us some hints to the small targets.
However, since a Weiner filter is costly in time, and also by using several fast Fourier transforms, the frames were shifted by 1 or 2 pixels diagonally, which makes it difficult to map the final result to the original frame. Thus, we replace the Weiner filter with a Gaussian filter with the same purpose of smoothing our images.

2.1.2 Detection state

In the first step of Detection, the processed image is feed through 2 modules: BSM and LCV. Both aims to eliminate false targets such as strong edges, clutter noises, etc. Then the image is combined to create a saliency map where we can easily threshold into a final mask.
An example of saliency map is in figure 5, with the small target having an incredibly high value, thus can be extracted using a threshold function.

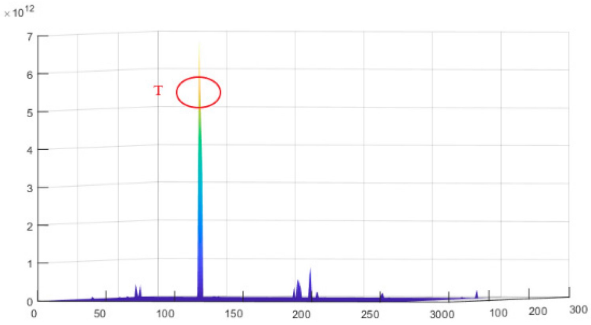

Figure 5: Final Saliency Map.

With our dataset, the BSM and LCV modules were not efficient enough to extract all noises. Therefore, a threshold is applied to each module to eliminate as many noises as possible. Also, instead of using mean value of the saliency map to be the threshold function, a ratio of 0.3 of the maximum value of the saliency map is used to determine the target points.

2.1.3 Properties of the model

The non-deep-learning model uses rather simple kernels to process the image, but can also provide a good result without any training steps. However it is really inconsistent: structure noises such as light signals from the ground can make it fail. Also, it provides a large number of false alarms which makes it inconvenient for tracking the target.

2.2 A modified version of the simple and efficient network for small target detection

2.2.1 The simple and efficient network for small target detection

In 2019, "a simple and efficient network for small target detection" has been proposed (Figure 6). One of the authors' main contributions was expanding the receptive field, and utilizing contextual and location information for small targets using some specific modules, of which we will explain the main points below. [7]

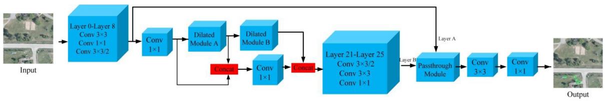

Figure 6: The proposed model. [7]

We will first explain the two "dilated modules" and the "passthrough module" of this network in general before proposing our modified version of it.

2.2.2 Dilated module

Dilated convolution A dilated convolution is a two-dimensional convolution layer with spacing between kernel points [8], as demonstrated in Figure 7 with dilated rate of 1, 2, and 4. Dilated convolution can help expand the receptive field exponentially without losing information, which is extremely important for small target detection [7].

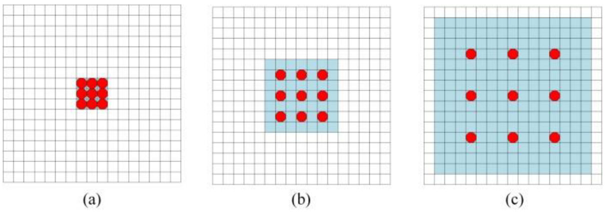

Figure 7: Dilated convolution with (a) dilated rate=1, (b) dilated rate=2, and (c) dilated rate=4. [7]

Dilated module In the feed-forward phase, a dilated module passes the input to a dilated convolution, then continues to reduce the dimension of the module by passing the received tensor to a 1x1 convolution. After that, we concatenate both the original input and the output tensor of the 1x1 convolution, then pass it to another 1x1 convolution to get the output of the module. The processes of the 2-dilated module and 4-dilated module are shown in the diagram in Figure 8. [7]

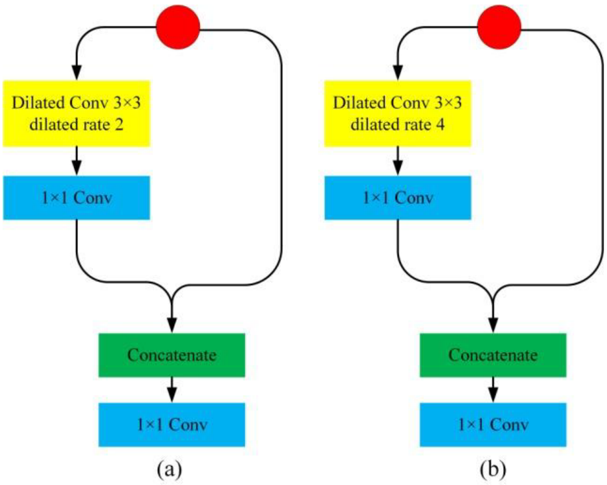

Figure 8: Dilated modules: (a) module A with a 2-dilated convolution, and (b) module B with a 4-dilated convolution. [7]

2.2.3 Passthrough module

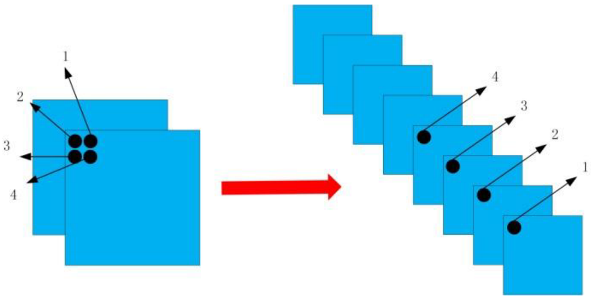

Figure 9: The implementation of passthrough layer. [7]

Passthrough layer The feature extracted from the previous layer can be effectively used by using a passthrough layer [7]. Specifically, in this model and our model as well, with a stride parameter of 2, it quadruples the number of channels and reduces the feature maps' sizes in each dimension by half. How the passthrough layer passes the input is demonstrated in Figure 9.

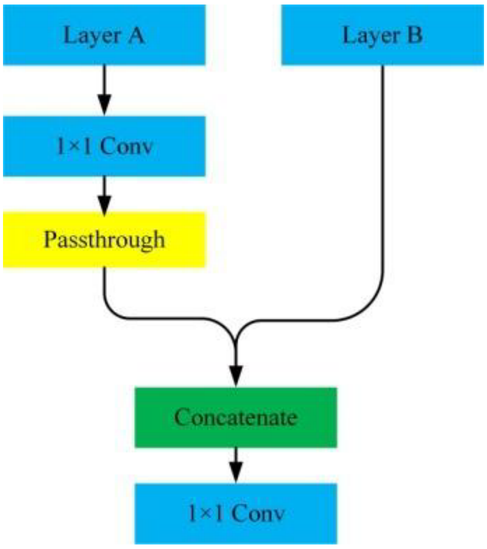

Figure 10: Passthrough module. [7]

Passthrough module As shown in Figure 10, the passthrough module gets two inputs from two different layers. The first input (whose feature maps' sizes in each of the two dimensions are double those of the second input) goes through a 1x1 convolution and the passthrough layer, then is concatenated with the second input. We will then feed the concatenated tensor to another 1x1 convolution to get the output of the passthrough module.

2.2.4 Training setup and observation

Our model, named Simpeff, is a slightly modified version of the proposed model we discussed at the start of this subsection (2.2.1). In particular, at the end of the proposed model, we added a global average pooling layer, followed by a flattening layer to connect it with a fully connected layer whose activation function is sigmoid. We also changed the image dimensions in all layers to fit our data instead of theirs. The diagram below (Figure 11) is our model architecture.

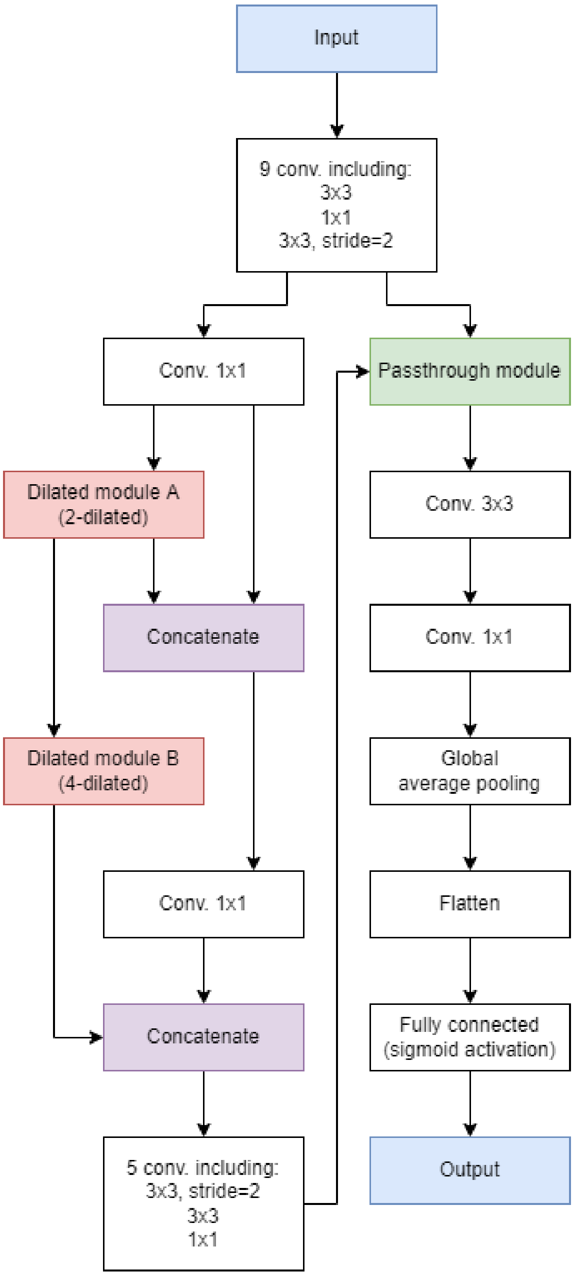

Figure 11: Simpeff architecture.

As the model has a task that are localization (regression problem), the intersection over union (IoU) loss variant - DIoU is selected as the overall loss for training. The optimization algorithm used is an adaptive learning rate method - Adam with default Pytorch hyperparameters.

IoU loss, which equals to $1 - IoU$, is preferable over L_n norm loss which are common for regression task since it is straight forwards to the purpose of object detection problem and also has many vantage points over the former [16, 17]. However, IoU loss only addressed the overlapped bounding boxes and will not provide any learning for the non-overlapping cases. To address this, researchers then proposed an improvement in the form of additional penalty term to the loss; it is called Generalized IoU Loss (GIoU) [18]. However, it will fall to the same case with standard IoU loss in the case of enclosing bounding box. This phenomenon is found and explained by Zheng et al. [19] in which they propose two better versions of IoU loss called Distance-IoU Loss (DIoU) and Complete-IoU Loss (CIoU).

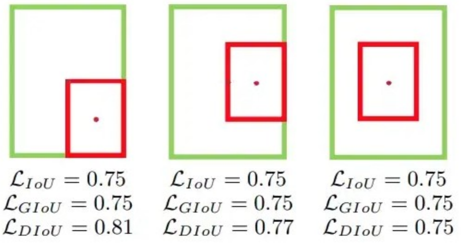
$\mathcal{L}_{IoU} = 0.75$
$\mathcal{L}_{GIoU} = 0.75$
$\mathcal{L}_{DIoU} = 0.81$
$\mathcal{L}_{IoU} = 0.75$
$\mathcal{L}_{GIoU} = 0.75$
$\mathcal{L}_{DIoU} = 0.77$
$\mathcal{L}_{IoU} = 0.75$
$\mathcal{L}_{GIoU} = 0.75$
$\mathcal{L}_{DIoU} = 0.75$

Figure 12: GIoU loss degrades to IoU loss for cases with enclosing bounding boxes, while DIoU loss is still distinguishable. Green and red denote target box and predicted box respectively. [19]

CIoU seems to be more promising than DIoU, according to Zheng et al [19]. However, in our case, DIoU possibly shines since one unexpected behaviour of CIoU increases the similarity of the aspect ratio, while it hinders the model from reducing the true discrepancy [20]. Unfortunately, our dataset has many equal-sized ground-truths.

The general trend of the training loss is decreasing, starting at about 1.8 and continuing to decrease to a certain loss of about 1.1, and at certain times, as a result of randomness, it fluctuated at that level (see Figure 13).

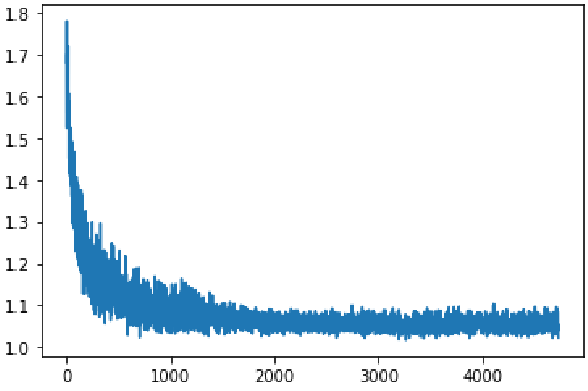

Figure 13: The training loss curve.

2.3 Infrared small target detection based on corner detection and convolution neural network

The architecture is composed of two modules. The first module, based on corner detection algorithm, takes an infrared image of any size as input. It then proposes multiple different regions of a fixed size (ROIs), each of which contains and centered around the suspected target points found by corner detector. Since there are many distinct types of corner (e.g., edge, flat, ...), we want the next module to classify whether a ROI contains an actual target but not an edge of random objects for instance, then localize more closely if there is at least one inside.

The inputs of the second module are the potential target regions (ROIs) forwarded by the lower layer. Each ROI is fed into a convolution neural network (CNN) to obtain the probabilities of whether the corner inside each ROI is a desired target. The category of the region depends on the maximum probability.

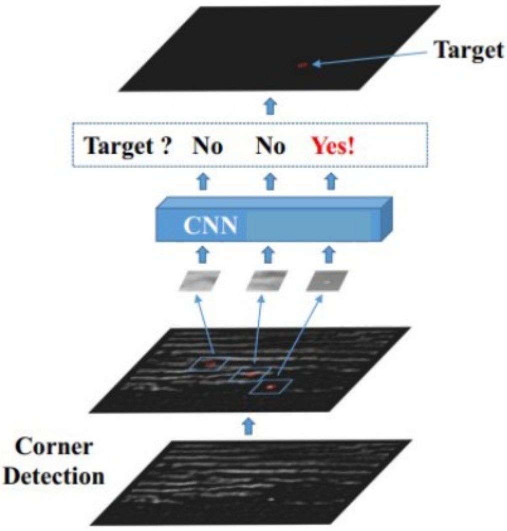

Figure 14: Corner detection based CNN framework overview. [9]

2.3.1 Corner detection

A corner is a point around which the pixels vary greatly in gradient amplitude and direction. Qi et al. proposed that the shape of small infrared target is similar to the isotropic Gaussian intensity function due to the optics point spread function (PSF) of the thermal imaging system at a long distance [10]. Figure 15 illustrates five typical examples and their gradient vector graphs. Hence, infrared small target is also a kind of corner.

Naturally, human intelligence has the ability to separate objects from the background across a frame, then detect them easily. One secrete may be the corners our vision recognizes that helps us to filter instances. The machine vision can also works similarly.

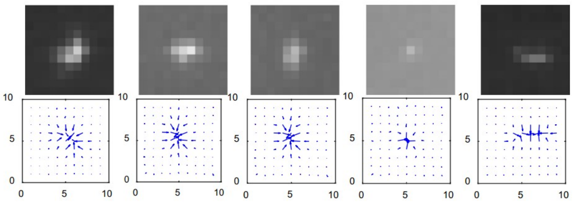

Figure 15: Gradient vector graphs of infrared small targets. The arrows indicate the amplitude and direction of the gradient. [9]

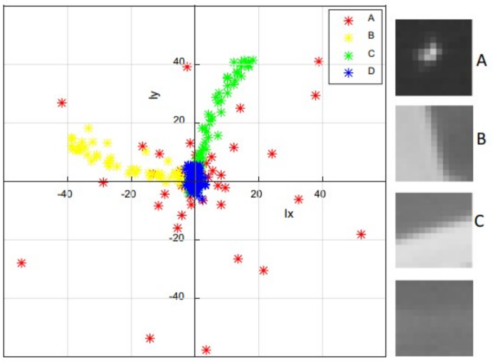

Figure 16: Gradient distribution of different regions. Each '*' represents the gradient of a pixel, the abscissa represents the horizontal gradient, and the ordinate represents the vertical gradient (A: target region; B, C: edge region; D: flat region). [9]

As the eigenvalues of the covariance matrix depend on the distribution of gradients, they can be used to distinguish targets from other regions. Therefore, we study the gradient patterns of the corners at first. Out of small point target, there are also flat regions (e.g., sky background,...) and edge regions (e.g., cloud clutter,...) . One very interesting feature is that the gradient distributions of such regions own visually different characteristics, as shown in the figure 16. This means there are places for us to filter out the corners that are desired targets.

From figure 16, the flat regions have small and multi-direction gradients distributed near the origin. The gradients of the edge regions point to a certain direction, which is dependent to the its surface. On the contrary, the gradients of the corner are scattered across the area, not limited to any direction and amplitude. As the result, if a region is flat-typed, the values of λ_1 and λ_2 are both small; if it is an edge region, then $\lambda_1 \gg \lambda_2$ or vice versa; and if it is a target. both λ_1 and λ_2 are large. Considering the complexity of eigenvalue calculation, the determinant and trace of the covariance matrix are used to measure eigenvalues. The process of such corner detection method is sum up into Shi-Tomasi corner detection algorithm [11], which works effectively for not only our scene but also many other backgrounds.

2.3.2 Feature extraction (CNN)

To eliminate the false target and locate more exactly the true targets inside ROIs, a CNN - pretrained Resnet-34 is selected to determine whether the candidate region contains the real target or not.

One of the issue with the lightweight CNN classifier of the original paper observed during training seems to be the vanishing gradient problem, shown in the figure 17. Also, since we want our model to localize closely the targets inside ROIs but not just assume the ROI as a legit bounding box for the target; therefore, a new architecture for CNN block is required. Furthermore, RCNN-like structure possibly takes a lot of time for training. These reasons then lead us to Resnet architecture (Resnet-34) with a wide range of pretrained checkpoints, that potentially tackles all sorts of issues above with skip connection mechanism and residual learning [12].

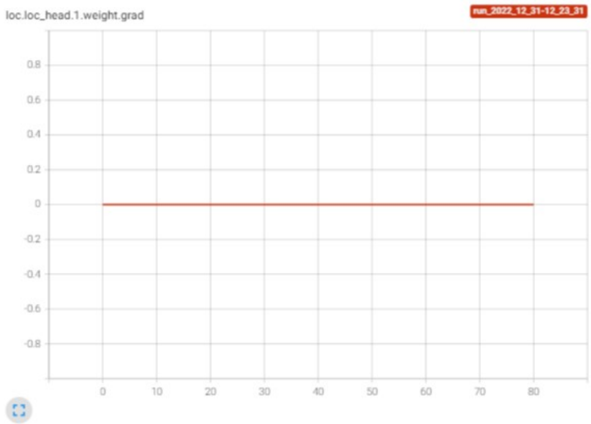

Figure 17: One of the layers of the original CNN [9] have zero gradients during training.

Resnet-34 consists of 3 residual units of 64 feature maps, 4 residual units of 128 feature maps, 6 residual units of 256 feature maps and 3 residual units of 512 feature maps. The head is made of a global average pooling layer and a fully connected layer of 1000 neurons. Since our problem just demands 4 coordinates and 1 score, then we replace the fully connected layer with two multi-layer perceptron (MLP) with corresponding number of outputs for each task (one is classification and four regression). Interestingly, the chosen network, Resnet-34, still includes pooling layers, maximum and global average pooling in particular. Intuitively, such pooling techniques are destructive, which means they drops lots of input values, hence causing the loss of weak signals that possibly include our small targets. However, since CNN just has to work with ROIs smaller than the whole frame and the size of targets compared with that of the ROI is possibly recognized, the pooling layers is a savior to reduce the training load as well as retaining the gray feature of the infrared small targets. Likewise, the global average pooling, selected as output, is beneficial to avoid overfitting (no parameter to be optimized) and especially sum out the spatial information that will improve the robustness against disrupted inputs.

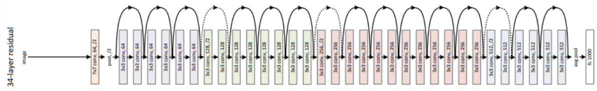

Figure 18: Resnet-34 architecture (with original upper layers). [12]

Last but not least, before forwarding ROIs to the CNN block, contrast adjustment of a fixed factor is applied on-the-fly. In this way, the signal of the target can be enhanced out of the visually noisy background (figure 24).

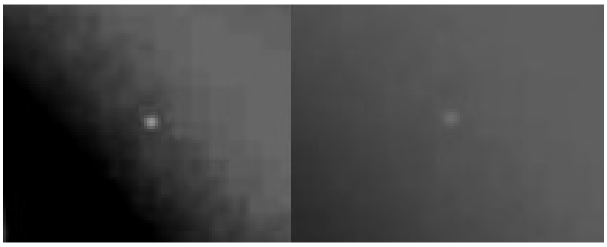

Figure 19: A contrast enhanced (left) vs. original ROI.

2.3.3 Training setting and observation

As the model has two tasks that are target classification and bounding box regression, the binary cross-entropy loss and intersection over union (IoU) loss variant - DIoU is selected to partially sum up the overall loss for training. The optimization algorithm used is an adaptive learning rate method - NAdam with default Pytorch hyperparameters. Since the resource is limited, the model is trained with batches of 32 instances, 1180 iterations per epoch, 20 epochs in total.

We train the model as a whole, with Resnet being frozen in few first epochs; till we observe fluctuation of the loss, which is the sign of that current model cannot learn more, then few next below layers will be opened up for fine-tuning. The final result is better with a possibly converged training loss curve, compared to the former model; however the validation raises the sign of overfitting. At last, we still expect something more.

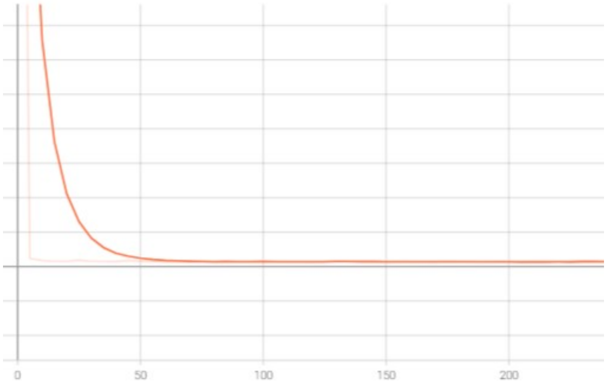

Figure 20: The training loss curve of the model.

2.4 A segmentation approach

Apart from detection models, segmentation models were also used widely. They are considered an upgrade of detection models as every single pixel are evaluated to determine whether it belongs to the object. For segmentation models, we use a MobileNetv3 encoder [13], as our problem is applied in military, which requires minimal devices. Two structures, UNet and Feature Pyramid Network (FPN) were experimented.

2.4.1 Dice loss

Small target detection problems suffer from a huge problem of class imbalance between our desired target and background, therefore Dice loss is suitable for our training process. The formula for Dice loss is

$$1 - 2 * \frac{y \cap y_{\text{pred}}}{y + y_{\text{pred}}}$$

2.4.2 UNet

UNet or U-shaped network is a powerful segmentation model. Originally introduced to medical imaging tasks [14], UNet has become one of the most used models related to semantic segmentation tasks. An example of UNet is portrayed in the following figure

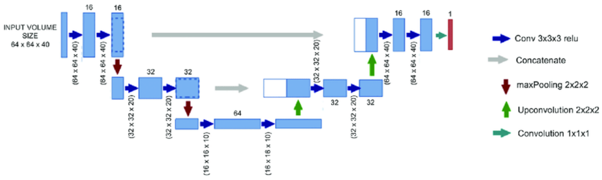

Figure 21: UNet architecture

So, why UNet? It is mainly because, for upper layers, they use valuable information of our target by concatenating the image itself with the feature map. Also, the process of downsampling and upsampling contributes greatly to addressing the target's coordinates.

For each layer of the encoder, the image size is halved. Therefore, only a 3-layer encoder is necessary for our model (in contrast to a 5-layer original UNet), since a small interpolation can make the target disappear from the image, making the training step unnecessary. Therefore, our UNet becomes even more lightweight. From our experiment, the UNet provided one of the best results using less than 200,000 parameters.

2.4.3 Feature Pyramid Network (FPN)

Originally, FPN [15] was built for object detection tasks. Its performance on small target was much greater than other models of the same category, therefore it becomes a candidate for solving our problem. This is derived from the fact that FPN uses a top-down approach, while the others use a bottom up approach. A brief look at the FPN model will explain more clearly why it performs better than many object detection models for this particular task.

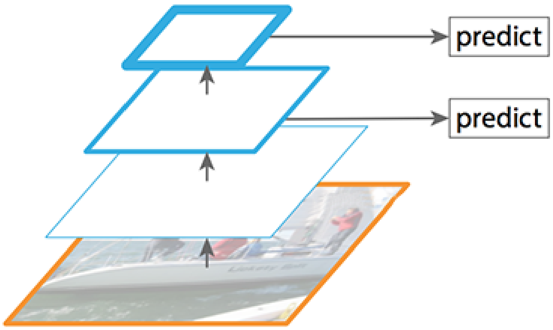

Figure 22: Bottom-up approach

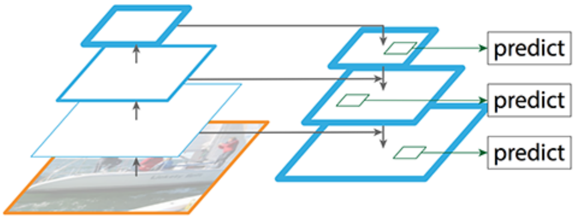

Figure 23: Top-down approach

For bottom-up approaches, the size of feature maps becomes smaller through layers, thus sometimes accidentally nullify the existence of the targets. However, with FPN, skip connections were used to protect the information from larger feature maps. Therefore, when applied as a segmentation model, FPN can perform well on our dataset. However, it faces a problem with huge parameter size, leading to slower inference time.

3 Experiment

3.1 Data collection and processing

Our dataset was collected in Infrared mode of Viettel High Tech's Electro-optic center's Observatory application. In our 60 collected samples, 30 were from 02 August, 2022 and the observe angle are larger than 45 degrees. Meanwhile, the rest of our dataset were collected on 10 August, 2022, on a lower angle and, apart from clouds, multiple buildings and roads had also been observed.

Objects were synthesized by selecting a random coordinate of the first frame of each sample, and then increase their contrast value by at least 10. Then the coordinates of the objects where shifted randomly from 1 to 3 pixels from itself of previous frame. The size of object remains constant for each sample, and is populated as in the following figure.

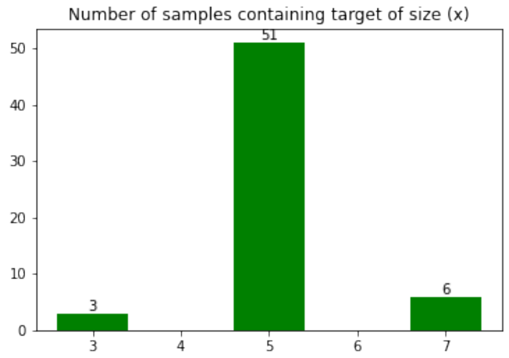

Figure 24: Major number of targets are of the size 5x5, over 60 distinct types of target.

The coordinates of object of each frame is logged onto a csv file for every sample. Thus we need to split frames of each sample into singular photos, and then make a new csv file that covers all samples for ground truth. Moreover, mask images were generated for our segmentation models. The split for training and testing were 90:10 as we want to maximize our value, and also the whole dataset is large enough to prevent overfitting, thus allowing such split.

3.2 Model evaluation

3.2.1 Evaluation metrics

In order to objectively evaluate the target detection performance of different methods, several widely used metrics in this area including detection rate P_d and false alarm rate F_a are introduced to our work.

$$P_d = \frac{\text{number of true target}}{\text{number of ground-truths}} \tag{1}$$

$$F_a = \frac{\text{number of false target}}{\text{number of detections}} \tag{2}$$

The detection rate P_d describes the sensitivity of a model to desired targets. In case of classification, P_d shares many characteristics with recall, and similarly the higher P_d is, the more real targets are detected. On the contrary, the false alarm rate F_a acts like the false negative rate (FPR), measuring the the model's ability to recognize whether a suspect object is truly a target. The higher F_a is, the less likely the model distinguishes the real targets from the false ones.

A prediction is count as true positive whenever its overlapping area with any ground-truth surpasses a fixed threshold. We set that threshold as IoU and raise three different values which are "> 0", "> 0.5" and "> 1.0" to consider whether the model have predicted the desired targets. Our aim is then to improve P_d and decrease F_a for all of our methods, across the three IoU thresholds.

3.2.2 Comparison and analysis

After computing the P_d and F_a on each of the above thresholds, the average detection rates, the average false alarm rates, and the perfect detection rates (P_d in $IoU = 1.0$) is calculated to compare the models (table 1).

Model	Avg. P_d	Avg. F_a	Perf. P_d (IoU=1)
WLCV (fr cut)	0.9156	2.4805	-
WLCV	0.4279	2.3453	-
Simpeff	0.1841	0.8159	0.0000
CCNN	0.3832	0.6796	0.0025
UNet	0.9843	0.0014	0.9293
FPN	0.9883	0.0014	0.5437

Table 1: Comparison across average detection rate (AP_d), average false alarm rate (AF_a), perfect detection rate (PP_d) for models.

From the result table, we can see that both WLCV models were falsely detecting many targets, while only the frame cut version managed to achieve a respectable detection rate of 91.56%. This model, however, was surpassed by both U-Net and FPN models, both in average detection rate and especially false-alarm rate. U-Net model managed to perfectly detect targets more accurately, with 92.93% compared to the 54.37% of FPN, despite the latter having slightly better detection rate. Simpeff and CCNN models are on the lower end of statistics, having next to no perfect detections and 18.41% and 38.32% detection rate, respectively. All in all, the U-Net model yielded the best result overall, with a close runner-up in detection rate, small false-alarm rate, and top-end perfect-detection rate.

4 Conclusion

In this paper, we propose several methods for solving the SIRST detection problem, which involves locating a small target on an infrared image. In particular, we used the method of weighted local coefficients of variables, a modified version of the simple and efficient network for small target detection, a model based on corner detection and convolutional neural network, U-Net, and the feature pyramid network. After evaluating them, we yielded the best model, which is U-Net. While it only has the second highest average detection rate of 0.9843, slightly lower than the highest of 0.9883, it has the highest perfect detection rate of 0.9293 compared to 0.5437 for the second highest, along with a very low average false alarm rate. For future work, we will continue to optimize the proposed models and look for new interesting methods to work on this problem of detecting small targets on infrared images.

References

[1] Chuanyun Wang and Shiyin Qin. “Adaptive detection method of infrared small target based on target-background separation via robust principal component analysis”. In: *Infrared Physics Technology* 69 (2015), pp. 123–135. ISSN: 1350-4495. DOI: `https://doi.org/10.1016/j.infrared.2015.01.017`. URL: `https://www.sciencedirect.com/science/article/pii/S1350449515000286`.

[2] Suyog D. Deshpande et al. “Max-mean and max-median filters for detection of small targets”. In: *Signal and Data Processing of Small Targets 1999.* Ed. by Oliver E. Drummond. Vol. 3809. International Society for Optics and Photonics. SPIE, 1999, pp. 74–83. DOI: `10.1117/`

12.364049. URL: https://doi.org/10.1117/12.364049.

[3] Bin Ye and Jiaxiong Peng. “Application of order morphology filtering on detection of small target and point target”. In: *Object Detection, Classification, and Tracking Technologies.* Ed. by Jun Shen, Sharatchandra Pankanti, and Runsheng Wang. Vol. 4554. International Society for Optics and Photonics. SPIE, 2001, pp. 94–99. DOI: 10.1117/12.441630. URL: https://doi.org/10.1117/12.441630.

[4] Lizhen Deng et al. “Adaptive Top-Hat Filter Based on Quantum Genetic Algorithm for Infrared Small Target Detection”. In: *Multimedia Tools Appl.* 77.9 (May 2018), pp. 10539–10551. ISSN: 1380-7501. DOI: 10.1007/s11042-017-4592-2. URL: https://doi.org/10.1007/s11042-017-4592-2.

[5] Lizhen Deng et al. “Adaptive top-hat filter based on quantum genetic algorithm for infrared small target detection”. In: *Multimedia Tools and Applications* 77.9 (May 2018), pp. 10539–10551. ISSN: 1573-7721. DOI: 10.1007/s11042-017-4592-2. URL: https://doi.org/10.1007/s11042-017-4592-2.

[6] Junmin Rao et al. “Infrared Small Target Detection Based on Weighted Local Coefficient of Variation Measure”. In: *Sensors* 22.9 (2022), p. 3462.

[7] Moran Ju et al. “A Simple and Efficient Network for Small Target Detection”. In: *IEEE Access* 7 (2019), pp. 85771–85781. DOI: 10.1109/ACCESS.2019.2924960.

[8] *Conv2d — PyTorch 1.13 documentation.* https://pytorch.org/docs/stable/generated/torch.nn.Conv2d.html. (Accessed on 01/01/2023).

[9] Mingming Fan et al. “Infrared small target detection based on region proposal and CNN classifier”. In: *Signal, Image and Video Processing* 15 (2021), pp. 1927–1936.

[10] Shengxiang Qi et al. “A Robust Directional Saliency-Based Method for Infrared Small-Target Detection Under Various Complex Backgrounds”. In: *Geoscience and Remote Sensing Letters, IEEE* 10 (May 2013), pp. 495–499. DOI: 10.1109/LGRS.2012.2211094.

[11] Jianbo Shi and Tomasi. “Good features to track”. In: *1994 Proceedings of IEEE Conference on Computer Vision and Pattern Recognition.* 1994, pp. 593–600. DOI: 10.1109/CVPR.1994.323794.

[12] Kaiming He et al. *Deep Residual Learning for Image Recognition.* 2015. DOI: 10.48550/ARXIV.1512.03385. URL: https://arxiv.org/abs/1512.03385.

[13] Andrew Howard et al. “Searching for mobilenetv3”. In: *Proceedings of the IEEE/CVF international conference on computer vision.* 2019, pp. 1314–1324.

[14] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. “U-net: Convolutional networks for biomedical image segmentation”. In: *International Conference on Medical image computing and computer-assisted intervention.* Springer. 2015, pp. 234–241.

[15] Tsung-Yi Lin et al. “Feature pyramid networks for object detection”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition.* 2017, pp. 2117–2125.

[16] Hamid Rezatofighi et al. “Generalized Intersection Over Union: A Metric and a Loss for Bounding Box Regression”. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition*

(CVPR). 2019, pp. 658–666. DOI: 10.1109/CVPR.2019.00075.

[17] Jiahui Yu et al. In: ACM, Oct. 2016. DOI: 10.1145/2964284.2967274. URL: https://doi.org/10.1145%2F2964284.2967274.

[18] Hamid Rezatofighi et al. “Generalized Intersection over Union”. In: (June 2019).

[19] Zhaohui Zheng et al. “Distance-IoU Loss: Faster and Better Learning for Bounding Box Regression”. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 34.07 (Apr. 2020), pp. 12993–13000. DOI: 10.1609/aaai.v34i07.6999. URL: https://ojs.aaai.org/index.php/AAAI/article/view/6999.

[20] Yi-Fan Zhang et al. *Focal and Efficient IOU Loss for Accurate Bounding Box Regression.* 2021. DOI: 10.48550/ARXIV.2101.08158. URL: https://arxiv.org/abs/2101.08158.