International Innovative Research Journal of Engineering and Technology
Vol. 7, Issue. 3, March 2022, pp. 01~17
ISSN: 2456-1983, DOI: 10.32595/iirjet.org/v7i3.2022.153

Dim Target Object Detection Using Deep Learning

Weldekiros Misgana Desalegne[1], Rui Qing Wu[2]
[1]BSc., Electronic Information Engineering, University of Electronic Science and Technology, China
[2]Associate Professor, School of Information and Communication Enginnering,
University of Electronic Science and Technology, China
[1]misgana21son@gmail.com, [2]rqwu@uestc.edu.cn

Article Info

Article history:

Received Sep 9, 2021
Revised Nov 15, 2021
Accepted Jan 11, 2022

Keywords:

Infrared(IR)/thermal image
Dim target
Object detection
Region based convolutional neural network(R-CNN)

ABSTRACT

With significant advancements in computer vision technology using deep learning algorithms, object detection has shown terrific performance in bright and clear targets. However, due to the low signal-to-noise ratio (SNR), the challenges of detecting and segmenting objects in dim and dark targets continue to affect computer vision applications in dark and visually polluted conditions. Dim object detection using infrared imaging based on a deep learning algorithm, Convolutional Neural Network(CNN) for IR images, is proposed to counter this challenge. By studying and analyzing a series of CNN algorithms, the author presents an application based on Mask R-CNN that is better in precision and speed to detect and segment target objects in infrared images—dim and dark targets.

Corresponding Author:

Rui Qing Wu,
Associate Professor, School of Information and Communication Enginnering,
University of Electronic Science and Technology, China
Email: rqwu@uestc.edu.cn

1. INTRODUCTION

1.1. Research Background and Significance

One of the challenges in computer vision applications is that real-life images are not always of the same type, size, and quality. For example, a self-driving vehicle needs to observe and continuously analyze its environment. However, sometimes there can be places with fog, rain, or the absence of enough light along the street, so the images in these circumstances are always dim. In addition to this, medical images, night camera images, and most satellite images are dim. Therefore, we propose this research to help create a general approach to deal with dim images, specifically detecting objects and instant segmentation of objects in dim targets.

With the continuously improving infrared imaging technologies, infrared/thermal cameras can capture more spatial information than ordinary RGB cameras in dim targets, dark places, targets with extensive light scattering, snow, and rainy conditions. Because infrared imaging has the characteristics of strong anti-interference ability, wide detection range, high positioning precision, strong camouflage target recognition ability, and long-distance detection. [1] For this reason, computer vision applications that deal with dim and dark conditions use images captured by infrared/thermal cameras.

Infrared (IR) imaging technologies have been used for military, astronomy, industrial and research settings for decades. [2] Recently, with the integration of 2D array chips [3], they have become more reliable and easily affordable for mass production and versatility. Mainly night-time pedestrian and vehicle detection [4] and farming crops and animals inspection applications can use FLIR cameras installed in a plain view or

drones. In addition to autonomous driving vehicles and industrial applications, infrared imaging and detection can also help wildlife conservation. For example, the rhino numbers in Kruger National Park of South Africa have dropped by about 70 percent during the past decade, mainly because of poaching and its knock-on effects. [5] This wildlife endangering activity is not limited to rhinos in South Africa but prevalent worldwide to various endangered wild animals. The viable solution can be to install day and night surveillance systems equipped with infrared imaging technologies in the national parks. Because many of these poaching activities were happening at night, having a precise and fast object detection and tracking application for an infrared image is vital, not only to detect poaching activities but also to inspect and track the animals' activities and situations.

There are many object detection algorithms for an infrared image using different classical machine learning algorithms, like top-hat transformation [6], the local greyscale probability distribution-based algorithm [7], singular value decomposition(SVD) [8], the principal component analysis (PCA)-based algorithm [9], and maximum-mean or maximum-median filtering [10]. On the other hand, feature extraction algorithms and classifiers, like histograms of oriented gradients (HoG) [11] scale-invariant feature transform (SIFT)-like oriented features[12], and support vector machines (SVM) [13] are frequently used. Researchers have also used an object detection method based on Haar-like features with boosted cascade classifiers trained for human detection in infrared images in several studies [4]. Even though these classical detection algorithms show good detection performance in simple features and proper feature extraction techniques, they comprise complicated calculations and have low detection performance in dim and ambiguous images [14][15]. They also have limitations on accuracy and detecting multiclass objects in an image [16]. Deep learning-based approaches, such as convolutional neural networks (CNN), can enhance object detection performance in infrared images with adequate training data. Several studies reveal that CNN-based object detection algorithms perform better than SVM-based classification approaches [14][15][17]–[20].

Furthermore, by adding a region proposal network(RPN), CNN is modified to RCNN[21], fast RCNN[22], and faster RCNN[23]. In addition, the author has researched regression/classification-based detection algorithms like YOLO[24] and SSD[25], which are easy to train and fast detectors but have limitations on dim image detection accuracy due to frequent localization errors. Faster R-CNN has a profound deep-learning ability and efficiency in detecting and classifying multiclass objects with complex spatial features[18]. It contains a region proposal network (RPN) after the last convolutional layer. This network can look at the last convolutional feature map, produce region proposals, and use the same pipeline as R-CNN, including ROI pooling, FC, and then classification and regression heads.

Although faster R-CNN achieved a breakthrough in detecting an object of interest in good quality images, it has limitations due to misalignment of pixels during ROI pooling and low frames per second (FPS). Mask R-CNN[26] is an extension to Faster R-CNN, replacing ROI pooling by ROI aligning and adding another fully connected network for instant segmentation. ROI aligning will avoid misaligning features caused by manipulations in ROI pooling. The author proposed an object detection and segmentation algorithm for infrared/thermal images based on a Mask R-CNN using the FLIR dataset.

1.2. State of Arts

Deep learning approaches dominate recent object detection applications. Methods such as YOLO, Faster, R-CNN, SSD4 are up-to-the-minute choices because of their efficiency[27]. For multiclass object detection in thermal images, YOLOV3-SPP-thermal[28] [29] using spatial pyramid pooling is the state-of-the-art holder. A paper[30] about nighttime person detection using faster RCNN is also the latest work on object detection in infrared images. However, it is a single class and is very slow to be applied for real-time object detection.

1.3. Contents and Innovation of the Thesis

This thesis presents a state-of-the-art object detection algorithm for dim targets and nighttime applications using a deep learning algorithm based on a detailed observation of several infrared object detection algorithms. The thesis uses Mask R-CNN-based image detection and classification for more than 80 classes based on a COCO data set training for a Mask R-CNN. Although the method can detect and classify 80 classes, there are only three to four classes in the FLIR dataset for transfer learning to detect objects of the target class at the highest accuracy.

1.4. Outline of the Thesis

In section 2, basic concepts of object detection and some insight about region-based conventional networks(R-CNN) are elaborated.

In section 3, a Basic understanding of infrared imaging and its application, including some techniques of enhancing images, are depicted.

In section 4, Mask R-CNN and its implementation for infrared image object detection are outlined.

In section 5, Experimental procedures, implementations are explained, and conclusions are drawn.

2. BASIC CONCEPTS

This chapter discusses the general concepts and ideas of deep learning algorithms related to image processing and computer vision algorithms used in this thesis.

2.1. Convolutional Neural Network (CNN)

Convolutional neural network (CNN) architecture is a type of deep learning model inspired by the organization of the animal visual cortex, where neurons are interconnected with other neurons in which one neuron can activate another neuron[31]. It is designed to adaptively learn spatial features in an image by representing each pixel/group of pixels of an image as a single entity/tensor in the input layer and taking into a series of convolution and pooling layers for feature extraction and a fully connected layer for final regression or classification.

At the end of each convolutional layer, there is a typical activation function called Rectified Linear Unit(ReLu). Convolutional neural network architectures with ReLUs train several times faster than other nonlinear activation functions[33].

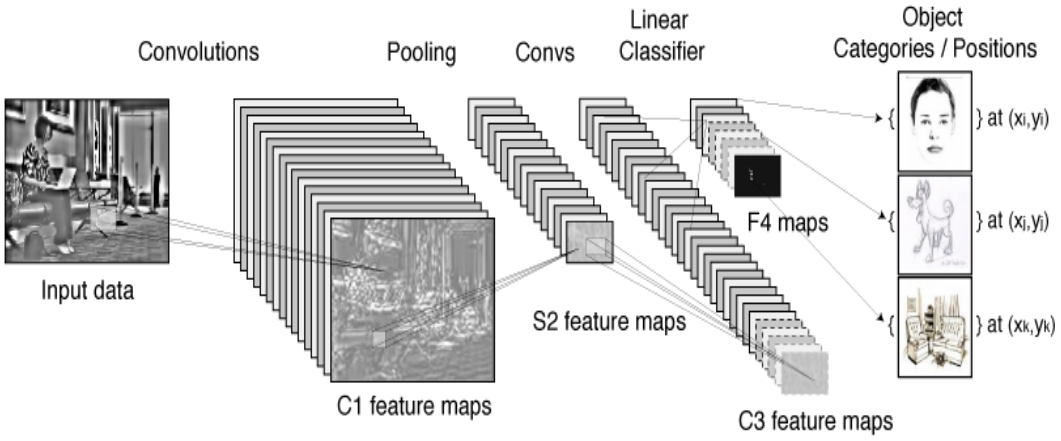


Figure 2-1. An overview of a convolutional neural network (CNN) architecture and the training process. figure[32].

2.1.1. Convolutional Layer

The Convolution layer is a series of convolutions with an associated kernel matrix multiplied with input tensors element-wise and added to give an output entity/ output tensor. The kernel sweeps across the image/matric and forms a feature map. This operation is repeated with multiple different kernels to create distinct feature maps.

The convolution expressed mathematically is $s(t) = (x * w)(t)$, where x(t) is the input function, and w(t) is the weight of the kernel. When we extend this in a matrix, form,

$$x \epsilon R^{H*W} \text{ and } w \epsilon R^{H\prime*W\prime}$$

$$S_{i,j,} = \sum_{m=-\left|\frac{H\prime}{2}\right|}^{\left|\frac{H\prime}{2}\right|} \sum_{n=-\left|\frac{W\prime}{2}\right|}^{\left|\frac{W\prime}{2}\right|} x_{i+m,j+n} w_{m,n}, \quad stride = 1 \qquad (2\text{-}1)$$

The hyperparameters that define the process are the size of the kernel matric and the number of kernels. In addition, the output feature can also vary based on padding and strides.

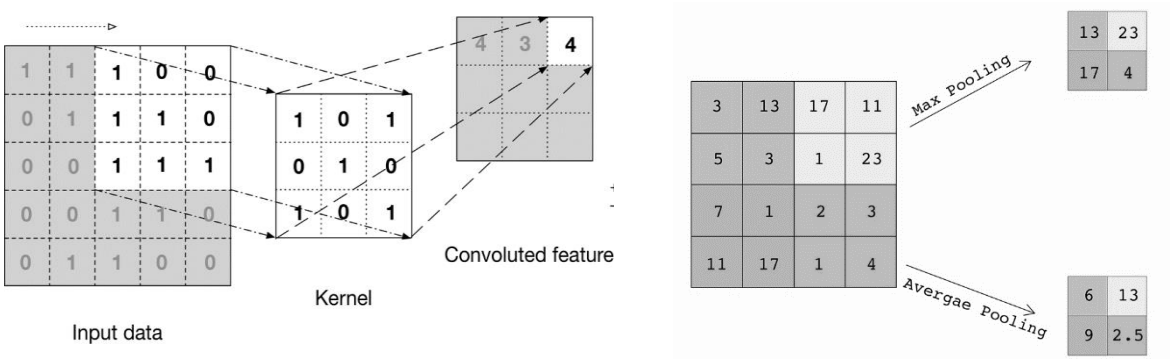


Figure 2-2. a) Convolution operation with stride 1, the input x and output s will have the same size.
b) Pooling layer, max pooling, and average pooling. Conceptual depiction.

The kernel matrix elements are weights that are training parameters. For example, if we have an image of 256*256 and a kernel of 3*3 with no padding and stride 1, we will have 3*3 = 9 parameters + 1 bias term. So if we have a total of 50 different kernels(filters), 50(3*3+1)= 500 parameters. If we have another 50 filters in the next convolutional, the total parameters in two layers will be 500(50(3*3+1))= 250,000 parameters. Learning is then varying the value of these parameters and keeping the ones that resulted in a better result. Hyperparameters like kernel size, padding, and stride can also be trained.

2.1.2. ReLU Activation Function

An activation function is a transformation function that changes the summed weighted input into an activation value for the node. A typical activation function that is a default activation function for many neural network algorithms is a Rectified nonlinear activation function. Because it helps to counter the problems of the linear rectifier and nonlinear rectifier, by acting as a linear, it makes it easy to train and easy for backpropagation. It also has a nonlinear property which makes it better to train complex features into deeper layers of the network. Deep convolutional neural networks with ReLUs train several times faster than their equivalents with *tanh* units[33].

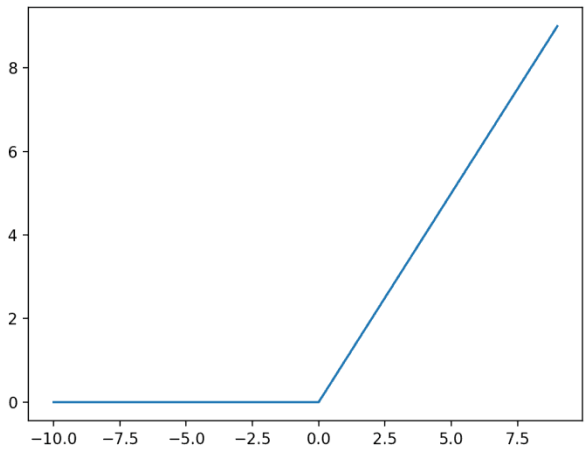


Figure 2-3. ReLU activation function $g(z) = \{^{0,\ z<0}_{z,\ z\geq 0}$

2.1.3. Pooling Layers

The first convolutional layer in CNN extracts feature maps and learns some low-level features after the activation function. Then a pooling layer is added to reduce the size and represent the feature maps with another fewer feature maps. It also helps prevent overfitting. Similar to convolution, pooling also has strides and paddings. However, the operation is not matrix multiplication but averaging the values or choosing the maximum value in that kernel. The kernel is not a filter in this case. There is no kernel matrix/ weights

involved in the pooling layer, so there is no parameter. Shown in Figure 2-2, b is a simple depiction of the pooling operation.

2.1.4. Fully Connected (FC) layer

In the Fully Connected layer, each neuron in the input connects to all the neurons in the output. After the final convolution and pooling layers, the feature map representation gets flattened into a one-dimensional array. Then the flattened 1D array is fed to a series of fully connected layers, all associated with learnable weights. The output of this layer is probabilities for each class.

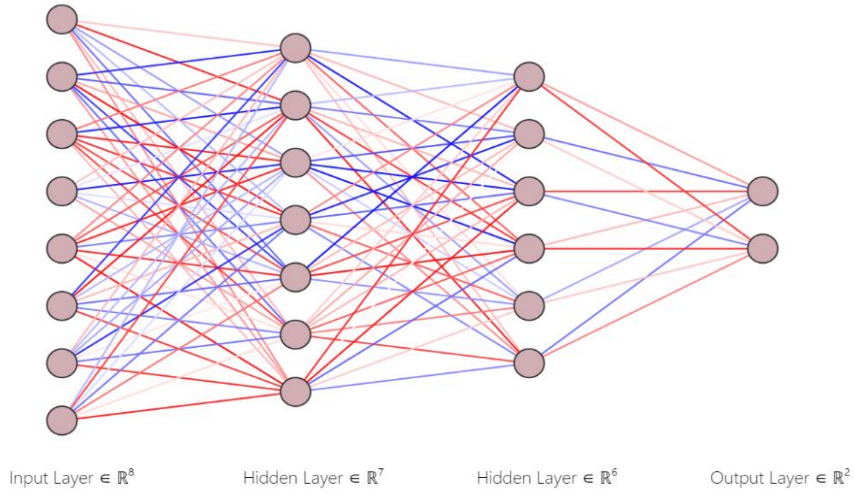


Figure 2-4. A typical representation of FC

2.1.5. Loss functions

A loss function, also referred to as a cost function, measures the compatibility between output predictions of the network through forwarding propagation and given ground truth labels. A commonly used loss function for multiclass classification is cross-entropy, whereas regression of continuous values mainly uses mean squared error. A type of loss function is also one of the hyperparameters in CNN.

2.2. Region-Based Convolutional Neural Network(R-CNN)

One of the challenges in object detection and classification using classical algorithms is that real-life images contain different types of objects in size, spatial location and aspect ratio, and shape. In some cases, the target can be covering most of the image, while in other cases, the target may be tiny relative to the background. So CNN algorithm processes the entire image to detect all types of objects in the image even if most of the region is a background, resulting in a tremendous amount of computation. A region proposal algorithm proposed in the R-CNN algorithm suggests many boxes in the image and checks if any of these boxes contain any object to reduce the number of regions and minimize computation and look only on these regions can avoid vast areas of background images which is not most of the time a target. R-CNN uses a selective search algorithm to extract just 2000 regions from the image [34]. The 2000 randomly selected regions are rearranged into a square and forwarded to feature extraction conventional neural network. The CNN then produces a 4096-dimension feature vector and feeds it to an SVM to classify and detect each proposed region.

Additionally, it predicts four offset values to increase the precision of the bounding box by adjusting the bounding box. After predicting the likelihood of an object, the RCNN model might predict multiple bounding boxes for an object Non-max suppression avoids multiple boxes based on their confidence score. On the other hand, when there is a bounding box that bounds no object, a negative example, which can be prevented by including it in the training dataset during the process.

Although region-based CNN reduces much computation, it takes a considerable amount of time to train the network. It takes much time because there are 2000 regions that the network processes take around 40-50 seconds per image, making it impossible for real-life implementation. The CNN, SVM, and bounding box regressor combined working over 2000 regions causes training difficulty and takes a vast memory. In addition, the selective search algorithm is not a machine learning algorithm but a fixed algorithm that always selects the 2000 regions for any image using the same strategy. The fact that no learning is happening here

indicates that there could be many useless regions proposed among the 2000 regions, which leads to redundant calculations in the network.

2.3. Fast R-CNN

The main problem of R-CNN is that the CNN feature extractor, the SVM classifier, and the bounding box regressor models process all the 2000 proposed regions, which takes too much time. To overcome the problem of R-CNN, the authors in Fast R-CNN [22] improved the training model by combining the three independent models and making them a joint trained structure. This structure combines them into a single CNN forward pass to the whole image and output a feature matrix. This feature matrix contains all the regions of interest (ROI). ROI pooling layer recipes all the feature maps into a specific fixed size. It passes each region into a fully connected layer along with a softmax layer for identifying classes and a linear regression layer for generating and optimizing a bounding box for each target. Fast R-CNN is faster than R-CNN because it does not feed 2000 region proposals to the feature extraction network every time. The feature extraction operation takes the whole image once and generates a feature map. Still, the regions of interest are proposed using a selective search model, an independent model like in R-CNN. As observed from this experiment, Fast R-CN takes around 2 seconds per image, better than R-CNN but impractical for large datasets in real-life applications.

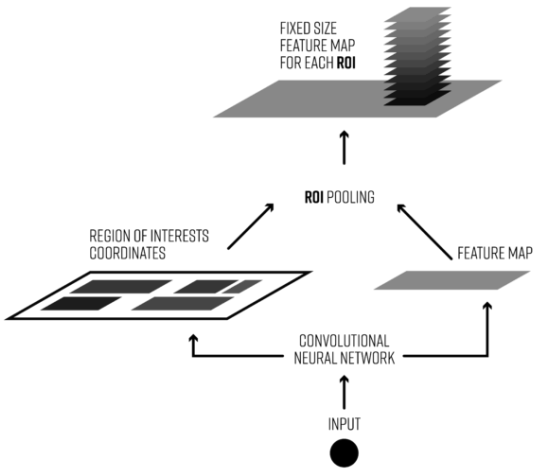


Figure 2-4. A Fast R-CNN model, including ROI pooling layer. The CNN outputs the feature maps, and a selective search proposes the regions. Then the ROI pooling layer reshapes all the feature maps into a fixed feature map for each ROI. (Image: Tomasz Grel)

2.4. Faster R-CNN

Like Fast R-CNN, Faster R-CNN contains a pre-trained feature extraction CNN network. Additionally, it has a Region proposal Network (RPN) to generate object proposals and another class predicting network. The RPN is inserted at the last conventional layer of feature extraction. This RPN model is trained to create candidate regions directly without using classical mechanisms like selective search. After the RPN layer, the network is the same as Fast R-CNN. We use ROI pooling, classifier, and bounding box regressor. First, an input image goes to the CNN network for feature extraction and returns with the feature map for that image. Then we apply RPN on these feature maps to produce object proposals and score for each proposal that indicates its confidence. Next, by using ROI pooling on each proposal, the model converts them into a fixed size compatible with the next step. These stacked proposals then go to a fully connected layer with a softmax layer for classifying the object and a linear regression layer at the top for boundary boxes. Another special thing about Faster R-CNN is that it also contains anchor boxes for capturing objects of different shapes and sizes based on the training dataset.

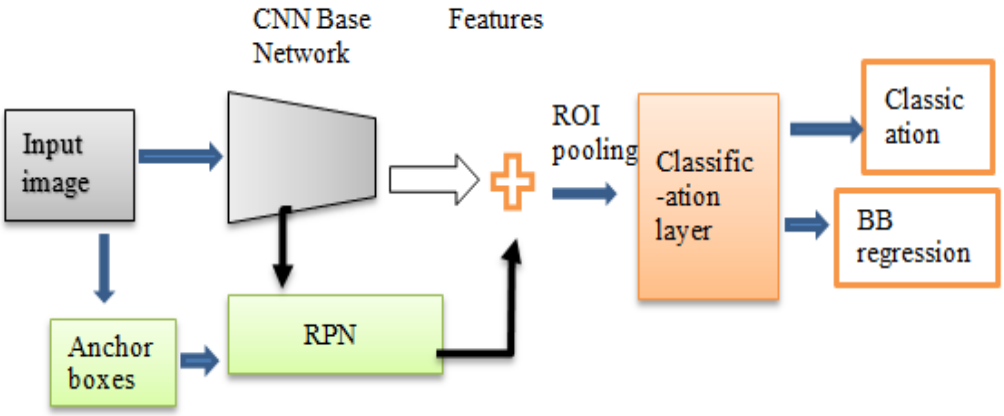


Figure 2-4. A Faster R-CNN model.

The CNN base network can be VGG-16, ResNet, or MobileNet. In this case, it is ResNet. RPN uses anchor boxes and its internal network to propose a region of interest. The features from the feature extraction layer are of different shapes and sizes. ROI pooling layer converts them to a fixed compatibility size.

2.3.1 Anchor Boxes

Anchor Boxes are different in size and aspect ratio bounding boxes placed uniformly in the whole area of the original image. After a pre-trained network identifies the location of possible color distinction or edge distinction, anchor boxes are assigned with different sizes and shapes (aspect ratio) to represent objects of varying shape. We use anchor boxes to evaluate all object's region proposals, multiple objects, objects of different shapes and scales and occulted objects without needing a sliding window scanner to detect by repeating the same procedure for each object class. In addition, anchor boxes are robust to translation to another location, and they can be used at every location in the image. The RPN network then receives the information in each anchor box. The regression in RPN gives offset values, and the classification layer identifies that each box contains an object or not by assigning a probability value. The correspondence property of CNN let extracted and passed through RPN figures in the anchor boxes can correspond back to their original location in the image. Anchor boxes are tunable parameters in the configuration of this model. For example, if there are sizes of 128*128, 256*256, and 512*512 and width to height ratios of 2, 0.5, and 1.5, then there will be a combination of 9 different anchor boxes for each anchor point.

2.3.2 Region Proposal Network(RPN)

A Region Proposal Network (RPN) is a trainable neural network model that takes an image feature of any size as an input and outputs a set of rectangular object proposals, each with an objectness score.[22] It is an essential element to bring the idea of an end-to-end deep learning algorithm into action for a faster image detection algorithm. After the backbone network(the CNNbase network) produces feature maps, they are represented by a corresponding anchor point. Each anchor point will contain all varieties of anchor boxes. The RPN determines if certain anchor boxes corresponding to each anchor include a target. The RPN classification layer determines the anchor box if it is background or a possible target region. Moreover, the regression layer calculates to optimize the coefficients of position, width, and height of the anchor box that contains the potential target region. To do this, the RPN model needs to train itself, classifying the areas included in each anchor box using layers of RPN classifier and RPN bounding box predictor. Anchor boxes are compared with ground truth based on intersection over unity(IOU). So we have two loss functions: RPN structure, Loss for training that classifier, and Loss for bounding box optimizer. When anchors overlap, non-maximum suppression(NMS) removes those anchor boxes with IOU scores above a predefined threshold.

2.5. Summary

CNN algorithm is a deep learning algorithm for image detection containing two main structures, feature extraction region and classification, and regression region. Although it can effectively detect objects in an image, it is slow to train and too sluggish for real-life implementation. Region-based CNN adds a region proposal layer to the CNN, making it effective in avoiding outside areas using a selective search algorithm to extract just 2000 possible regions of interest from the image. Extending the importance of

region-based CNN, Fast R-CNN further improves the mechanism of R-CNN by avoiding 2000 regions passing through the CNN layer. Instead, it generates the regions (2000-40000, based on the task) using a selective search and skips the feature extraction layer and joining the feature maps in the ROI pooling layer. As a continual, Faster RCNN avoids using a particular search algorithm for a region of interest proposal. Instead, it introduces a trainable region proposal network (RPN) that contains a classification and regression layer, which makes it a two-stage image detection algorithm.

3. OBJECT DETECTION APPROACHES IN INFRARED IMAGES

3.1. Introduction

Since its discovery, the "infrared rays"[35] were used mainly for thermal measurement. Thermal detectors continue to be invented and modified, primarily focusing on temperature measurement. Later the thermal sensors improved thermal detection with distance and extreme precision. Even after discovering thermal cameras, the applications of Infrared imaging technology had been limited for medical and military use only. However, with the improving infrared imaging technologies, infrared/thermal were found to be more applicable than ordinary RGB cameras in dim targets, dark places, targets with extensive light scattering, snow, and rainy conditions.

3.2. Infrared ray and Thermal Radiation

Black body radiation states that[36] 'All bodies with a temperature above absolute zero (-273.15 8C) emit thermal radiation as a function of their temperature.' For a black body, which considered a perfect emitter, spectral radiance is:

$$L_\lambda = {2hc^2}\Big/{(h^5\left(e^{{hc}/{\lambda kT}}-1\right))} \qquad (3\text{-}1)$$

L_λ Spectral radiance, also defined as radiance flux density that emits from the unit surface area, and h is plank's constant, k the Boltzmann's constant, and λ wavelength. However, based on the concept of emissivity, every natural surface is not a perfect emitter. The actual emitted radiant may be less than the theoretical black body's emission. The emissivity value is the ratio of the actual radiance of a surface to the theoretical black body radiance, so some materials are better emitters than others. The strength of the radiance(infrared waves) depends not only on the emissivity of the surface but also on the body's temperature. Things on the earth have different temperatures and different emissivity, and both independently affect the radiance of an object. It is crucial to separate the effects of temperature and emissivity on an object's radiance to study the factors independently. The radiance emitted from objects corresponds to the infrared lights in the electromagnetic spectrum.

3.3. Thermal Infrared Imaging and Image analysis in Thermal Imaging

3.3.1. Thermal Imaging

Thermal imaging or thermography is scanning surfaces by detecting and scanning the infrared rays emitted from every object in the target area. The thermal sensors (cameras) use robust temperature sensors. Since no external light is applied to shine a target area, all the light source is every object in the image, and infrared imaging is a passive technique. Because everything that emits infrared has different emissivity, the infrared cameras can distinguish individual objects in the image even if they have the same temperature

3.3.2. Thermal Imaging analysis

Thermal images are characterized by low resolution, and they need some image enhancement procedures before they are used for any case. They use image enhancement to minimize the random noise and cross-talk problems by improving the visual contrast of an image. Each pixel in the infrared image represents the corresponding temperature with a grayscale number, where the highest temperatures are characterized by white and the lowest temperatures with black. To reduce the image noise, low pass filters are applied, and to enhance edges, spatial derivatives of pixel intensity can be used.

Objects with similar property and same temperature with a background and other objects have a minimal difference in their radiance energy. As a result, the thermography will have invisible edges. However, the distance and location of such objects may vary that needs an edge representation in the image, but too small to be visible. We use thresholding to eliminate the background against a specific range of temperature values to counter such circumstances. So these seemingly edgeless distinctions will reveal their edge because the background becomes dark. Thresholding or removing background based on some pixel fixation is all that

can be done with single-channel thermal images. When additional spectral information is available, many alternative methods enhance and separate edges. A more sophisticated approach to eliminating irrelevant pixels is to use multispectral images and image analysis. This can be done by grouping pixels based on each represents temperature values and applying multiple thresholding to group the image into various instances and backgrounds.

3.4. Object detection Approaches for Infrared Images

Object detection in infrared is becoming more and more practical in applications such as early-warning, military, search and track, remote sensing, and medical imaging. The approaches to detect images comprise classical image detection algorithms to recently developed deep learning algorithms. 'Generally, a model of infrared image mainly contains three basic components: target, background, and noise.'[37]

$$f_O(x,y) = f_T(x,y) + f_B(x,y) + f_N(x,y) \quad (3\text{-}2)$$

Where f_O, f_T, f_B, f_N are actual image, background, target, and noise consecutively and (x,y) location of pixels in a two-dimensional image. In the case of targets far away relative to the size, the target becomes a circular spot which can be represented by the two dimensions Gaussian intensity model [38]

$$I_{target} = I_{max} \exp\left(-\frac{1}{2}\left[\left(\frac{x}{\delta_x}\right)^2 + \left(\frac{y}{\delta_y}\right)^2\right]\right) \quad (3\text{-}3)$$

The target is represented by I_{max} the peak intensity, δ_x and δ_y are two dimensions distribution parameters representing how the spot looks like, vertical ellipsoid, horizontal ellipsoid, or circle. Sometimes the target can be taken as a point target when it is too small, and simplified representation is needed. If we take the circular spot as a point target then, (x,y) $\rightarrow$ 0, I_{target}=I_{max}=1.

The simplest method to detect targets (large or small) is using simple thresholding, as discussed in section 3.3. For example, a drone in the sky can be stood apart by simple thresholding. We can avoid using a fixed threshold by using a Gaussian-weighted sum of local intensity values, also known as adaptive thresholding, which allows the thresholding procedure to be more robust to changes in lighting conditions. In addition, detecting moving objects relative to the background, the fact that the background remains still makes it possible to use background subtraction. The moving target can be seen from pixels changing over time compared to pixels that remain unchanged. If the detector is also moving, we can use the relative motion of the target to the background. Gaussian models represent the targets as equation (3-3) or using K-means clustering. Using K means clustering part of the image with intensity in a particular range is considered to represent the target.

The above-mentioned classical target extraction methods are not robust in complex scenarios. Infrared imaging in real life consists of complicated multiple objects and backgrounds. In addition, for applications in modern-day technology, they should detect and apply segmentation for a better representation of the environment. Precision and speed are the main goals in image detection and tracking models. The trend to create robust real-time image detection and instance segmentation using deep learning algorithms continues.

3.5. Summary

Black body radiation states that every material emits thermal energy as a function of its temperature. A thermal camera can detect the different radiant from any object represent it with a pixel intensity in an image, these creating an infrared/thermal image. These images are characterized by low resolution, and they need some image enhancement procedures before they are used for any case. The classical method to detect targets (large or small) in infrared images is simple thresholding, K-means clustering, adaptive thresholding, etc. Moving objects relative to the background can be detected easily by background subtraction. For robust real-time image detection and instance segmentation, deep learning algorithms are promising.

4. MASK R-CNN FOR DETECTION AND INSTANCE SEGMENTATION OF OBJECTS IN INFRARED IMAGES

4.1. Introduction

Mask R-CNN is one of the latest and influential object detection and instance segmentation deep learning model. As depicted in section 2.3, Faster R-CNN detects an object, classifies then outputs a bounding box and class label of the object. 'Mask R-CNN extends Faster R-CNN by adding a branch for predicting an object mask in parallel with the existing branch for bounding box regression[26]. Image segmentation is a crucial step for a clear representation of the environment in computer vision applications. Classical segmentation algorithms for infrared images are not robust to noise, complexity, and geometry. Nor are they speedy and accurate. Using Mask R-CNN as a detection and segmentation framework, we can

perform infrared image detection and segmentation. Dim target images—infrared images are characterized by generally low resolution and noise. Applying multiple convolutions and poolings can cause the image to lose most of the features, which causes a poor detection rate. However, with a particular backbone in the feature pyramid network (FPN)[39], we can preserve essential features in the pyramid while keeping the CNN operation as deep as possible. In addition, the ROI alignment procedure avoids the misalignment problems during ROI pooling which was one cause of distortion for shallow features.

4.2. Mask R-CNN For Dim Target Object Detection

Mask R-CNN contains two stages; the first one in the region proposal stage to generate possible target regions. The second is the classification and segmentation stage, where bounding boxes and masks are produced; thus, it is a two-stage image detection model. The two stages are connected to the backbone structure or a base CNN network. In Mask R-CNN, the Backbone is ResNet, specially structured based on a feature pyramid network(FPN). CNN base in the form of FPN maintains robust semantical features at various depths[39]. Shallow features and edges will be preserved at the early convolutions, and in-depth features will be detected at the deep convolutional layers. Another neural network in the second stage takes proposed regions by the first stage, assigns them to several specific areas of a feature map level, scans these areas, and generates objects classes, bounding boxes, and masks. In Figure 4-1, The dimension of the image can be any size of above 512. The Backbone(ResNet), FPN, RPN, and RoI Align are depicted clearly.

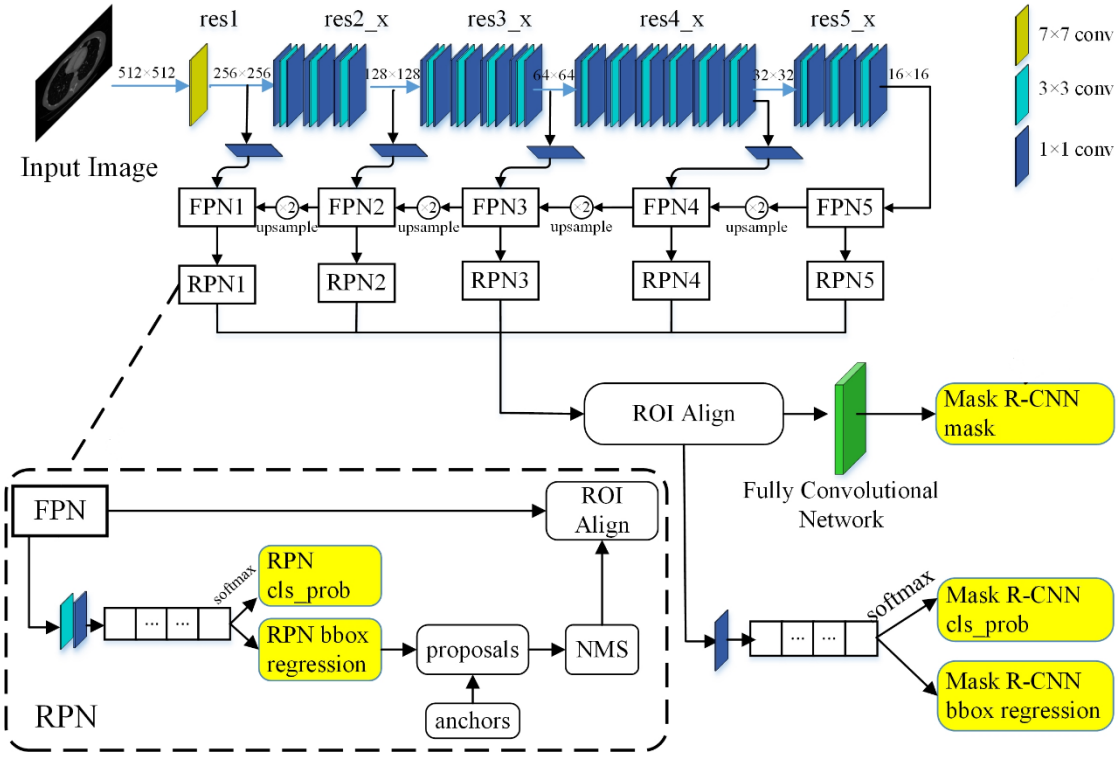


Figure 4-1. A generalized diagram of Mask R-CNN used in an infrared image. The Backbone(ResNet), FPN, RPN, and RoI Align are depicted clearly. Image source[40]

4.2.1. The Backbone Residual Network

The Mask R-CNN in this thesis uses ResNet50 as a backbone, a multilayer neural network that bring out the original image's feature maps. The backbone system does not require a training network with random initial parameters because it can use pre-trained parameters. Using transfer learning, one can efficiently train with fewer datasets and avoid overfitting. There are other backbone structures, such as VGG[41], ResNet, and DenseNet. Unlike ResNet, VGG does not have skip connections in its deep layers. Features in infrared images do not reach the desired depth. It also has more parameters than ResNet 50. As a result, ResNet 50 is a viable option as a backbone of this framework and using the COCO pre-trained parameters.

4.2.2. The Feature Pyramid Network(FPN)

Object detection in infrared images with complex features and scales is challenging, especially with far and small objects, for various reasons. For example, small objects and distance targets occupy a few pixels in the infrared image. The information in these few pixels can disappear or change in the continuous convolution and downsampling process. In addition, the images are characterized by low SNR; the edges that distinguish the foreground and background can vanish between the layers; thus, the feature map will miss important information about the image resulting in poor detection performance.

The solution is to apply ROI prediction at the end of every layer in the CNN base by passing the result of each layer to the Region proposal Network(RPN) using a single-scale feature map. Then the RPN can produce the region of interest (RoI)s for every layer. However, this can be time-consuming and memory extravagant; besides, the running is towards real-time object detection and instance segmentation.

Another solution is to use the same method as the above. Instead of taking the results of the convolutional layers, RPN takes features extracted from each of these layers using a feature pyramid network(FPN)[39]. Figure 4-1 shows that the FPN goes from right to left and opposite the backbone (top-down pathway). From the backbone network in ResNet, the FPN extracts feature maps from five different levels of the network to detect objects at different scales with different resolutions and contain different levels of features. Feature maps from early layers are higher resolution containing low-level semantic information, whereas the deeper feature maps contain low resolution and high-level semantic information. The shallow features(high resolution) are helpful to detect small objects, whereas deeper features(low resolution) are essential for complex objects. The FPN uses different feature maps and upsample and combines them in a top-down pathway (right-left in figure 4-1) with lateral connections or Figure 4-2 with an upsampling layer and a 1x1 convolutional layer.

4.2.3. The Region Proposal Network (RPN)

As depicted in figure 4-1, RPN takes every feature map in FPN one by one and predicts the region of interest in each feature map. By selecting 16 anchor boxes, the ROI of different scales will be aligned using the ROI Align operation. It's worth noting that the RoI Align step enhances RoI accuracy by using bilinear interpolation instead of the rounding procedure used by RoI pooling. It's worth noting that there will be a lot of anchors to choose from in this stage. Targets of various sizes correlate to different ratios and scales. According to data, pulmonary nodules range in size from 3 to 33 millimetres, thus the anchor and scale must be able to accept a wide range of nodule sizes and have a high need for identifying small nodules.

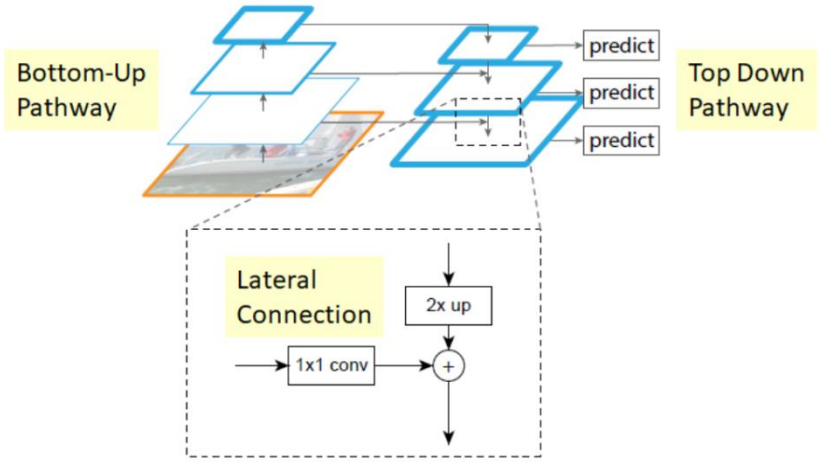


Figure 4-2. A simplified overview of the FPN working concept

4.2.4. The Region of Interest Alignment (RoI Align)

Another improvement to Faster R-CNN is the introduction of RoI Align to bring all the RoI proposals into a fixed size. The RPN has utilized different scale feature maps from different convolution levels. So The RoIs predicted by RPN are of different sizes and have different resolution and semantic information. To concatenate all the ROIs together, we need to bring them into a fixed size without losing essential information. In Faster R-CNN, this task is done by RoIPooling, whereas, In Mask R-CNN, it is done using RoIAlign. Unlike RoIPooling, RoIAlign is not characterized by quantization problems. It performs RoI mapping and pooling without quantization. It has four steps. The first is RoI mapping by multiplying the scaling factor to the corner coordinates of the RoI. Next is RoI division, dividing the height and the width of

the RoI into k by k grid without quantization. Then RoI interpolation, dividing each grid into four subgrids and finding centroids in each of the four subgrids, then finding their value using bilinear interpolation. And max pooling. Finally, applying max-pooling, one value from each of the four subgrids in all k*k. RoI Align creates a k*k feature map. As depicted in[26], Mask R-CNN using ResNet-50 and stide 32 performs better in segmentation and bounding box detection using RoI Align than RoI Pooling. As observed in the paper[26], misalignment is more severe with larger strides.

4.2.5. Functions

After the RoI Alignment, the thousands of feature maps are ready for classification, detection, and segmentation networks. These three function divisions are classification, detection, and segmentation. The classification branch will have a convolution layer and softmax. The detection branch will perform bounding box regression. The segmentation branch will be associated to a entirely convolutional layer, and each class in the infrared image will generate a binary mask.

Mask R-CNN adopts a two-stage image detection procedure. RPN stage and prediction class, boundary box, and binary Mask for each RoI. This approach is very similar to Fast R-CNN when it applies to the bounding box regression and classification in parallel. During training, we define a multi-task loss on each RoI as:

$$L(.) = L_{classification} + L_{b.box} + L_{mask} \quad (4\text{-}1)$$

The classification and the bounding box regression are the same as those in R-CNN. If we have k classes in our image, then the mask branch has a K m*m-dimension output for each RoI. Applying per pixel level activation function defines Lmask as an average binary cross-entropy loss. For RoI with a ground truth K, the mask is only defined on the Kth mask without competition with other classes.

5. EXPERIMENT PROCEDURE, RESULTS, AND CONCLUSIONS

5.1. Experiment Procedure

The MaskR-CNN with a ResNet100 was selected as the model architecture to be executed. First, it was trained with the 2016 COCO challenge dataset. Finally, Bayesian optimization was executed and verified to enhance the model's hyperparameters. All the networks were trained on a Google Colaboratory, and for assessment, the metric mAP was utilized with the norms IoU=0.5, the similar as the Pascal object detection challenge metric. During the training procedure, the Adams optimizer with learning rate = 10−5 and clipnorm = 10−3 was utilized. The training consisted of 50 epochs with a total of 10000 steps each epoch. On the Imagenet dataset, all trained models used a backbone network with pre-learned weights.

The ResNet with 101 layers (ResNet100) as the backbone network was implemented. On the Imagenet, the backbone network was used with pre-trained weights. To recognise objects in RGB images, the model was trained on the COCO dataset train set. The model was then tested using the mAP@IoU=0.5 measure on the validation set.

Because IR images have only one channel (luminance) but the RGB images have three (red, green, and blue), the author decided to produce three input images from the single IR image.

$$w_1 r + w_2 g + w_3 b \rightarrow (w_1 + w_2 + w_3) i$$

Where w1, w2 and w3, are the weights applied to the red, green and blue input channel, and i is the infrared input channel. If the infrared channel is used as all three input channels, this is true. The IR image was triplicated and used as input for all three channels, using the same model architecture as the one trained on the COCO dataset using RGB images.

Step By Step Experiment Procedure

To make easy inspection during training, the model contains three inspection modules so that it is possible to inspect the data, the model, and the weights at each point.

Anchor sorting and filtering

It inspects the steps at the first stage of RPN and visualizes the positive and negative anchors and the refinement process of the anchor box.

Bunding Box Refinement

In this module, the bounding box is refined during the process, and the final bounding boxes are displayed and inspected.

Mask Generation

After feature extraction and ROI alignment, the mask head generates a pixel-level segmentation mask covering the target object pixel-wise. The refinement process and the final mask are inspected and displayed. Then the resulted mask is scaled and placed on the image at the final output.

Figure 5.1 a) Generated mask for a person instance after some epochs. b) A layer activation result of a particular image.

Layer activations

The base network(ResNet101) in this deep neural network framework contains different layers with different depths. There are activation points at each layer, and inspection of the activations at different layers is helpful to look for signs of problems of random noises or all zero phenomena and debug before wasting time in a fruitless training. Figure 5-1 b.

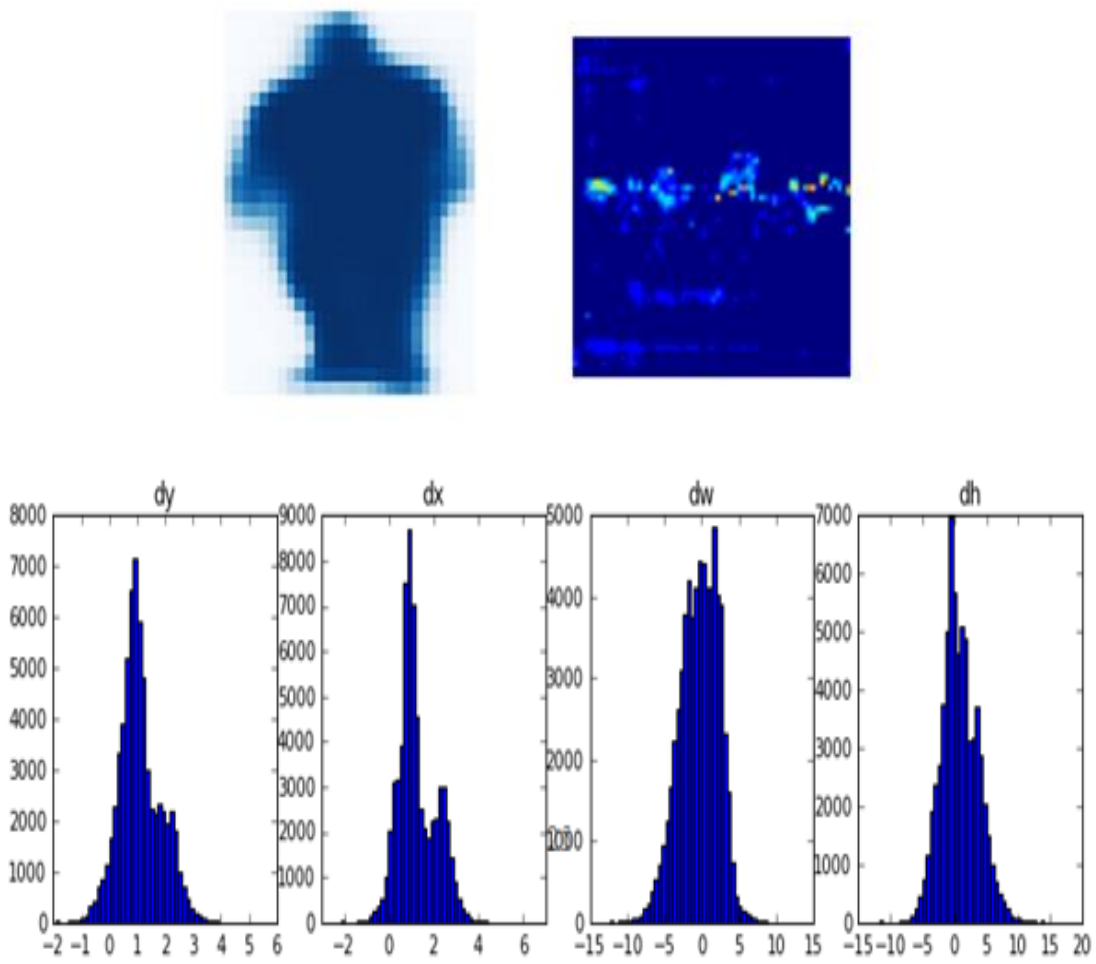
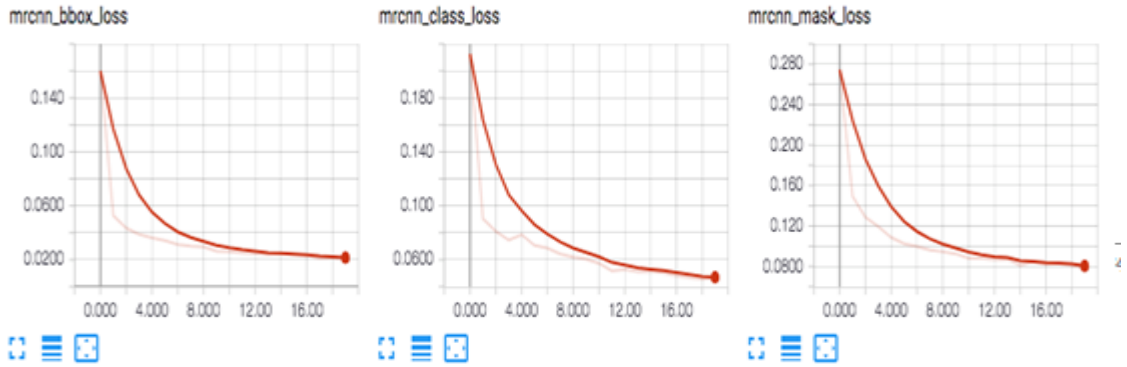


Figure 5.1. a) Generated mask for a person instance after some epochs.
b) A layer activation result of a particular image.



Weight Histograms

Since there are millions of weights in the Mask R-CNN model, aggregation is done using histogram statistics. It uses to find out how the aggregation of weights in the training process of the framework. So inspecting the weights using the histogram is another helpful tool during training.

Analyzing and Inspection the Loss Functions Using the TensorBoard

As discussed so far in this thesis, this specific image detection framework contains mainly three tasks. Object detection and localization using a bounding block regression, classification of the detected object and, mask producing for segmented instances. As a result, there are three loss functions are corresponding to each task. These loss functions are visualized using TensorBoard by analyzing the performance at the end of every epoch.

5.2. Results and Conclusions

The framework in this project was pre-trained in the RGB COCO dataset [42]. However, with little transfer learning from the FLIR dataset, the framework can detect and segment instances at the precision of 0.7 up to 1. The detection and segmentation speed is also improved. Generally, it takes below 200 milliseconds per image. Furthermore, it can detect up to 80 classes. Thus the model can achieve good detection performance without going expensive to get the training infrared dataset. However, to get a robust and well-optimized framework, a tailor-made dataset is crucial.

Figure 5-1. Detection and instance segmentation of dim target objects in infrared using Mask R-CNN

5.3. Discussions

The author demonstrated the advantage of using infrared images over RGB in dark and high light scattering conditions. There are two pictures taken at the same time and position, but different cameras, RGB camera, and Infrared imaging. The two images were concatenated or attached into one single frame and applied the image detection algorithm in this thesis. The conclusion can be inferred from Figure 5-2 that the infrared imaging is robust to darkness and light scattering; hence the objects can be detected using a deep learning algorithm in this thesis. This demonstration and conclusion also apply to foggy, snowy, smoky targets.

Another critical point as a discussion is for a specific task that does not need instance segmentation; we can use the characteristics that make Mask R-CNN better. What makes Mask R-CNN better in performance than Faster R-CNN is that the former uses FPN and RoI Align. Therefore, when the problem is focused only on object detection, we can model a bounding box detection framework using RPN and RoI Align to Faster R-CNN and avoiding the mask head, thus saving memory and reducing training time.

Figure 5-2. The top part of the image is RGB, while the bottom is an IR image of the same position and time.

5.4. Future Works

Implementing a fast and accurate real-time infrared object detection framework is crucial to avoid the dependency of computer vision applications on visible light. Utilizing the concept of FPN further and using a complete infrared dataset and using high-resolution cameras, it can be possible to create augmented reality at night by pixel level detecting everything.

ACKNOWLEDGMENTS

Upon completing this thesis, the author would like to thank Ruiqing Wu (Associate Professor) for his continuous support and motivation.

REFERENCES

[1] J. Du, H. Lu, M. Hu, L. Zhang, and X. Shen, “CNN-based infrared dim small target detection algorithm using target-oriented shallow-deep features and effective small anchor,” *IET Image Process.*, vol. 15, no. 1, pp. 1–15, 2021.
[2] H. Lee, “Handbook of Sensor Networking,” vol. 35, pp. 267–280, 2015.
[3] E. Bercier, P. Robert, D. Pochic, J.-L. Tissot, A. Arnaud, and J. J. Yon, “Far infrared imaging sensor for

mass production of night vision and pedestrian detection systems," in *Advanced Microsystems for Automotive Applications 2012*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 301–312.
[4] A. Gaszczak, T. P. Breckon, and J. Han, "Real-time people and vehicle detection from UAV imagery," in *Intelligent Robots and Computer Vision XXVIII: Algorithms and Techniques*, 2011.
[5] SANParks. South African National Parks Annual Report 2019-2020 [J], 1–128, 2020.
[6] M. Zeng, J. Li, and Z. Peng, "The design of Top-Hat morphological filter and application to infrared target detection," *Infrared Phys. Technol.*, vol. 48, no. 1, pp. 67–76, 2006.
[7] S. C. Xu Bin and W. Lian, *Dim targets detection based on local gray probability analysis*.
[8] H. L. Qin, H. X. Zhou, and L. Sq, "SVD for infrared dim and small target background suppression," *Semiconductor Optoelectronics*, vol. 30, pp. 473–476, 2009.
[9] C. Wang and S. Qin, "Adaptive detection method of infrared small target based on target-background separation via robust principal component analysis," *Infrared Phys. Technol.*, vol. 69, pp. 123–135, 2015.
[10] S. D. Deshpande, M. H. Er, R. Venkateswarlu, and P. Chan, "Max-mean and max-median filters for detection of small targets," in *Signal and Data Processing of Small Targets 1999*, 1999.
[11] A. D. Algarni, "Efficient object detection and classification of heat emitting objects from infrared images based on deep learning," *Multimed. Tools Appl.*, vol. 79, no. 19–20, pp. 13403–13426, 2020.
[12] S. Song, J. Zhu, X. Li, and Q. Huang, "Integrate MSRCR and mask R-CNN to recognize underwater creatures on small sample datasets," *IEEE Access*, vol. 8, pp. 172848–172858, 2020.
[13] F. Alsulami, K. Ahmadi, and E. Salari, "Detection and tracking of dim objects in infrared (IR) images using Support Vector Machine," in *2016 IEEE International Conference on Electro Information Technology (EIT)*, 2016.
[14] H. Qu, L. Zhang, X. Wu, X. He, X. Hu, and X. Wen, "Multiscale object detection in infrared streetscape images based on deep learning and instance level data augmentation," *Appl. Sci. (Basel)*, vol. 9, no. 3, p. 565, 2019.
[15] O. Yardimci and B. Ç. Ayyıldız, "Comparison of SVM and CNN classification methods for infrared target recognition," in *Automatic Target Recognition XXVIII*, 2018.
[16] A. Frome *et al.*, "Large-scale privacy protection in Google Street View," in *2009 IEEE 12th International Conference on Computer Vision*, 2009.
[17] M. Hasan, S. Ullah, M. J. Khan, and K. Khurshid, "Comparative analysis of svm, Ann and CNN for classifying vegetation species using hyperspectral thermal infrared data," *ISPRS - Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.*, vol. XLII-2/W13, pp. 1861–1868, 2019.
[18] Z.-Q. Zhao, P. Zheng, S.-T. Xu, and X. Wu, "Object detection with deep learning: A review," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 11, pp. 3212–3232, 2019.
[19] B. Khalid, A. M. Khan, M. U. Akram, and S. Batool, "Person detection by fusion of visible and thermal images using convolutional neural network," in *2019 2nd International Conference on Communication, Computing and Digital systems (C-CODE)*, 2019.
[20] J. Park, J. Chen, Y. K. Cho, D. Y. Kang, and B. J. Son, "CNN-based person detection using infrared images for night-time intrusion warning systems," *Sensors (Basel)*, vol. 20, no. 1, p. 34, 2019.
[21] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014.
[22] R. Girshick. Fast R-CNN[R]. 2015.
[23] E. Hanna and M. Cardillo, "Faster R-CNN2015," *Biological Conservation*, vol. 158, pp. 196–204, 2013.
[24] R. Huang, J. Pedoeem, and C. Chen, "YOLO-LITE: A real-time object detection algorithm optimized for non-GPU computers," in *2018 IEEE International Conference on Big Data (Big Data)*, 2018.
[25] W. Liu *et al.*, "SSD: Single Shot MultiBox Detector," in *Computer Vision – ECCV 2016*, Cham: Springer International Publishing, 2016, pp. 21–37.
[26] M. R-cnn, P. Doll, and R. Girshick. Mask R-CNN [J].
[27] J. Beyerer, M. Ruf, and C. Herrmann, "CNN-based thermal infrared person detection by domain adaptation," in *Autonomous Systems: Sensors, Vehicles, Security, and the Internet of Everything*, 2018.
[28] M. Zilkha and A. B. Spanier, "Real-time CNN-based object detection and classification for outdoor surveillance images: daytime and thermal," in *Artificial Intelligence and Machine Learning in Defense Applications*, 2019.
[29] Y.-Q. Huang, J.-C. Zheng, S.-D. Sun, C.-F. Yang, and J. Liu, "Optimized YOLOv3 algorithm and its application in traffic flow detections," *Appl. Sci. (Basel)*, vol. 10, no. 9, p. 3079, 2020.
[30] J. Park, J. Chen, Y. K. Cho, D. Y. Kang, and B. J. Son, "CNN-based person detection using infrared images for night-time intrusion warning systems," *Sensors (Basel)*, vol. 20, no. 1, p. 34, 2019.

[31] K. Fukushima and S. Miyake. 267–285, 1982.
[32] R. Yamashita, M. Nishio, R. K. G. Do, et al. Insights into Imaging, vol. 9 no. 4, 611–629, 01-Aug-2018.
[33] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, 2017.
[34] R. Girshick, J. Donahue, and T. Darrell, *Rich feature hierarchies for accurate object detection and semantic segmentation Tech report (v5)*.
[35] W. Herschel, "XIV. Experiments on the refrangibility of the invisible rays of the sun," *Philos. Trans. R. Soc. Lond.*, vol. 90, no. 0, pp. 284–292, 1800.
[36] H. G. Jones. 107–163, 2004.
[37] S. S. Rawat, S. K. Verma, and Y. Kumar, "Review on recent development in infrared small target detection algorithms," *Procedia Comput. Sci.*, vol. 167, pp. 2496–2505, 2020.
[38] K. L. Anderson and R. A. Iltis, "A tracking algorithm for infrared images based on reduced sufficient statistics," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 33, no. 2, pp. 464–472, 1997.
[39] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," *arXiv [cs.CV]*, 2016.
[40] L. Cai, T. Long, Y. Dai, and Y. Huang, "Mask R-CNN-based detection and segmentation for pulmonary nodule 3D visualization diagnosis," *IEEE Access*, vol. 8, pp. 44400–44409, 2020.
[41] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv [cs.CV]*, 2014.
[42] J. Du, H. Lu, M. Hu, L. Zhang, and X. Shen, "CNN-based infrared dim small target detection algorithm using target-oriented shallow-deep features and effective small anchor," *IET Image Process.*, vol. 15, no. 1, pp. 1–15, 2021.