Our framework is based on Double Multi-scale Feature Pyramid Network (DM-FPN), which utilizes inherent multi-scale pyramidal features and combines the strong-semantic, low-resolution features and the weak-semantic, high-resolution features simultaneously. DM-FPN consists of a multi-scale region proposal network and a multi-scale object detection network, these two modules share convolutional layers and can be trained end-to-end. We proposed several multi-scale training strategies to increase the diversity of training data and overcome the size restrictions of the input images. We also proposed multi-scale inference and adaptive categorical non-maximum suppression (ACNMS) strategies to promote detection performance, especially for small and dense objects. The overall structure of DM-FPN is shown in Figure 1:

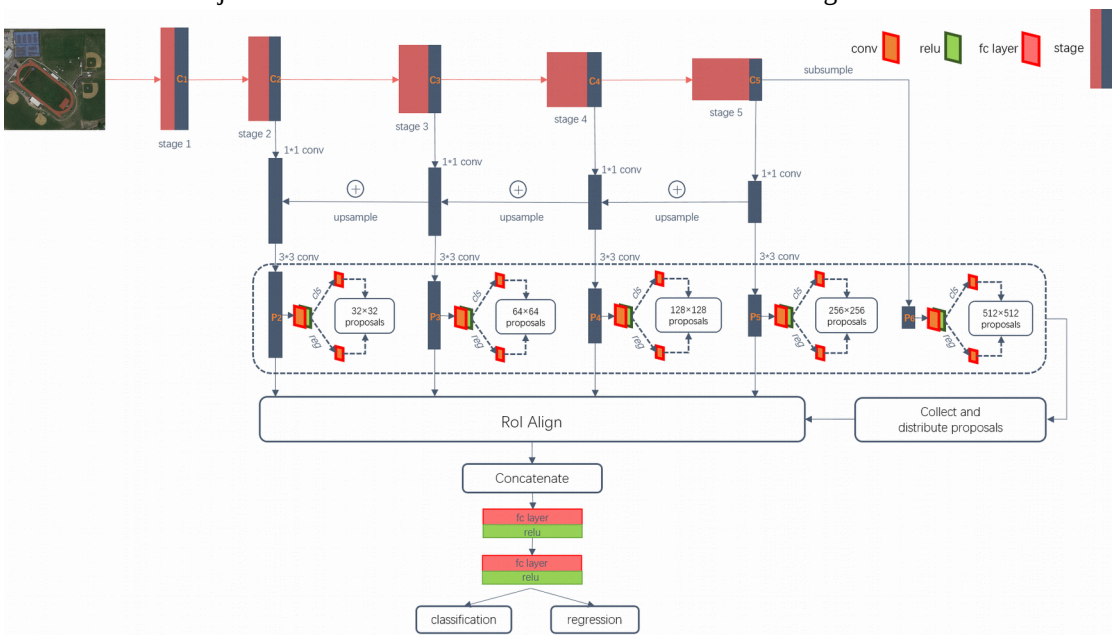

Figure 1. The overall structure of the proposed DM-FPN. It consists of a multi-scale region proposal

network and a multi-scale object detection network. These two modules share convolutional layers.

1. The infrastructure of DM-FPN DM-FPN

The infrastructure of DM-FPN is based on Faster R-CNN with FPN. Formally, both the original region proposal network and the detection network were modified by FPN. DM-FPN combines coarse-resolution, semantically strong features with high-resolution, semantically weak features, and such operations have great advantages in detecting small objects. We adopt ResNet50 as backbone of our framework. The convolution can be divided into 5 stages and the output of each stage's last residual block was selected as $\{C_2, C_3, C_4, C_5\}$, noting that they have strides of {4, 8, 16, 32} pixels corresponding to the original image. We do not utilize the first stage because it is memory-consuming. This process is called the bottom-up pathway. The corresponding $\{P_2, P_3, P_4, P_5\}$ were obtained by top-down path, lateral connections and mergence. Actually, to eliminate the aliasing effect of upsampling, a 3×3 convolution is executed on each merged feature map to obtain the final feature maps $\{P_2, P_3, P_4, P_5\}$, which are shared by the region proposal network and the class-specific detection network.

2. Multi-Scale Training Strategies

Multi-scale training strategies mainly include the patch-based multi-scale training data and the

multi-scale image sizes used during training. Their descriptions are as follows:

(1) Patch-based multi-scale training data. The size restrictions of the input images cause a lot of semantic information will lost in the deep convolutional layers, especially for small objects. Therefore, we slice remote sensing images into patches with a certain degree of overlap, and then send these image blocks into the network for training. At the same time, considering the uneven distribution of objects on the remote sensing image, which may include large objects such as playgrounds, and may also include small objects like cars, we enlarge and shrink remote sensing images by a factor of 2 and 0.5 respectively. The enlarged remote sensing images enhance the resolution features of the small objects while the shrunken remote sensing images integrally divide the large objects into a single patch for training.

(2) Multi-scale image sizes used during training. In order to enhance the diversity of objects, we adopt multiple scales for patches during training. Each scale is the pixel size of a patch's shortest side and the network uniformly select a scale for each training sample at random.

3. Multi-Scale Inference Strategies

We scale images to detect as many objects as possible during inference, and the scaled images include enlarged and shrunken images, horizontally and vertically flipped images. Specifically, we first perform multi-scale process on each test image, then we slice it into patches with a certain degree of overlap according to its size and carry out detection on these image blocks. Finally, we apply ACNMS to these concatenate bounding boxes from each patch to get the final results.

4. Adaptive Categorical Non-Maximum Suppression (ACNMS)

NMS is a post-processing module in the object detection framework, which is mainly used to delete highly redundant bounding boxes. A single remote sensing image may contain one big object or hundreds small objects, thus there exists a class imbalance between different categories. In the previous multi-class object detection works, the NMS thresholds for different categories are the same, but we find that different NMS thresholds for different categories based on the category intensity (CI) can improve the accuracy of object detection to a certain extent. We define CI as:

$$CI = N_{Ioc} / N_{img}$$

where N_{IoC} means the total number of instances for each category, N_{img} means the total number of images. If the CI of a category is greater than the given threshold, we set this category a larger NMS threshold than the generic NMS threshold. In general, NMS thresholds for denser objects are larger because they overlap each other more commonly.