Animal Detection in Huge Air-view Images using CNN-based Sliding Window

Young-Chul Yoon
School of Electrical Engineering and Computer Science
Gwangju Institute of Science and Technology
Gwangju, South Korea
zerometal9268@gist.ac.kr

Kuk-Jin Yoon
School of Electrical Engineering and Computer Science
Gwangju Institute of Science and Technology
Gwangju, South Korea
kjyoon@gist.ac.kr

***Abstract*—Our work concentrates on detecting tiny animals in huge air-view images with competitive accuracy and speed. Many environmental organizations investigate distribution of specific species of animals by capturing images from the sky. It is very challenging work for human to check the huge images and mark animals by hand. To check it automatically, we propose the method using CNN-based sliding window. There are popular works like Faster R-CNN or SSD that detect multiple objects in image. Despite their state-of-the-art performance, they are not applicable in this situation. Air-view image is huge and animals are tiny as not easy for human to detect. Normal multiple object detection methods are not suitable to detect tiny objects, which are smaller than minimum size threshold. Also, dataset is not suitable to train their networks. The ground-truth dataset doesn't contain scale information. In this paper, we introduce our own method, from training network using dotted ground truth dataset to detection and classification. Also, we verify the competitive performance of our multi-viewpoint based detection comparing with single viewpoint detection.**

Keywords—object detection; object classification; deep learning

I. INTRODUCTION

Image category classification is the conventional research topic in computer vision area. With the increasing popularity of the deep learning research, classification has been highly developed recently. State-of-the-art methods are mostly based on CNN(convolutional neural network). Popular CNN structures like VGG [1], GoogleNet [2], ResNet [3] have been proposed in ImageNet challenge. They are originally developed for classification work and have shown competitive performance. Because of its simplicity, VGG structure is frequently applied or embedded in other works as a feature extractor. We also adapted VGG-16 network in our method. Development of image classification inspired related works. Multiple object detection, combined with region proposal is one of them. Region proposal is necessary process in detection because of its ability of reducing size of problem domain. Without region proposal, we need to search whole image pixels with various size of windows. It takes huge time especially when processing big image. Many works have tried to combine region proposal with CNN based classifier [4] [5] [6]. Although, those works have shown competitive results in several benchmarks, like PASCAL VOC or ImageNet, they could be applied in specific domain. What if the training dataset consists of dotted images without any size information (See Figure 1 and Figure 2). Also, we need different approach since objects are ambiguous and too small to detect. To solve this problem, we propose the CNN-based sliding window method. The sliding window traverse the image and detect tiny sea lions. Our framework considered the speed and performance simultaneously. The detail is available in section 2 and 3. In my best knowledge, this is the first work that focus on detecting and classifying tiny animals in huge air-view image.

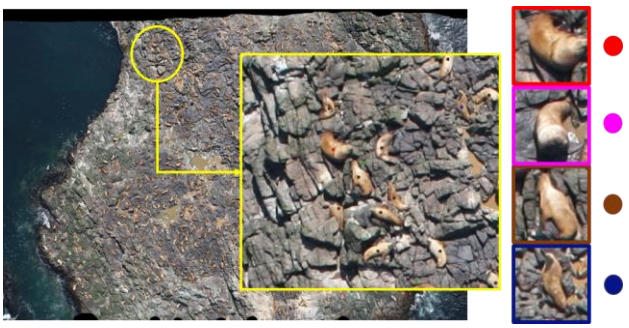

Fig. 1. Dotted ground truth image example. Each color of dots indicates specific age or sex of sea lion(red : adult male, magenta : subadult male, brown : female, dark blue : juvenile).

Fig. 2. Dataset doesn't provide bounding box information for each sea lion. Scale, proportion and orientation are all different for each sea lion and even many of them are overlapped. So, it's impossible to train and use popular networks like Faster-RCNN or SSD.

II. PROPOSED METHOD

Brief framework is introduced in Figure 3. Our framework is based on sliding window. The sliding window traverse the

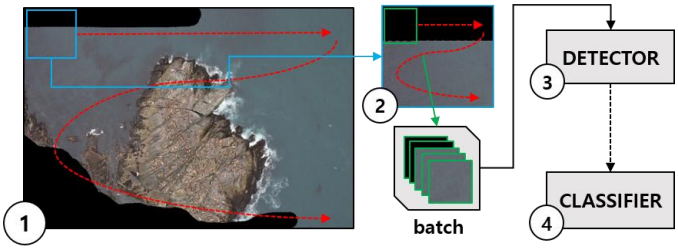

Fig. 3. Brief framework of our work. In each sky blue colored box area, green box traverses and collects patches as a batch. Using batch as an input of detector is much faster than one by one.

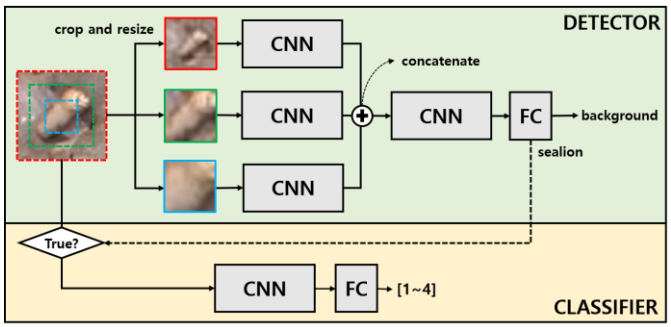

Fig. 4. Proposed network structure. Original patch size is 64x64 and is cropped and resized to 16x16. Because of same size of three inputs it can be processed in parallel and provide more information to dectector than only using one input.

image with 50% overlap and extract patches. Extracted patches are checked by CNN based detector and classifier. The classifier discriminates sex and age of sea lions(See the colored dots in Figure 1).

A. Detection and classification network

Our detection network considers speed and accuracy. As you can see in the Figure 4, original input patch(64x64) is cropped and resized into 3 images, which have different point of view. But these 3 images have same size 16x16. Top image has low resolution but wide sight. It contains rough and wide information. Bottom image has high resolution but very limited sight. It contains detailed and narrow information. We tried to apply the 3-level coarse-to-fine structure in our network. And because of their same size(16x16), 3 images are processed completely parallel without any bottle-neck. Finally, 3 types of features are concatenated and processed in remaining layers. The final output of detector network is an array, which consists of probability of sea lion and background. Patches, which have the sea lion probability over 0.5 are accepted as sea-lion. Detailed depiction of our network is drawn in Figure 5. Classification is similar to VGG-16 network. It was just downsized to fit to our patch size. The only patches, which contain sea lion, enter into classifier network. It reduces the time consumption by processing patches selectively.

B. Multi-level sliding window

Applying sliding window on whole image is quite slow. Especially, when each patch is processed by heavy CNN-based network, it takes enormous time. To reduce the time

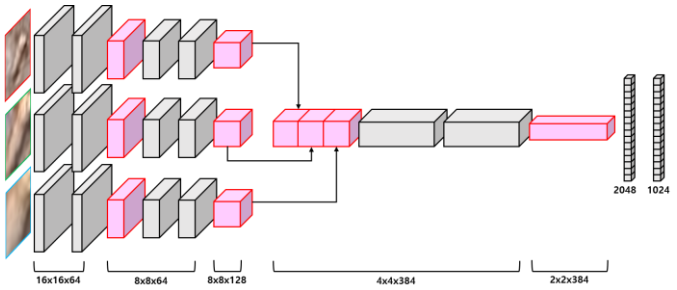

Fig. 5. Detector network. Magenta colored cube means max-pooling.

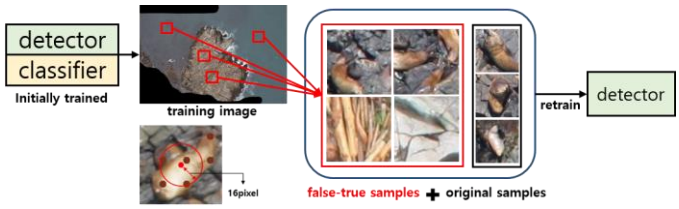

Fig. 6. Re-training framework. Apply initially trained detector and classifier on training images(0-749). And collect false-true samples which are further than 16-pixel from all ground-truth locations. We re-train our detector combining false-true samples and original samples.

consumption, we implemented multi-level sliding window. First, we divided the whole image into 64 sections. Then, took each section sequentially(Step 1 in Figure 3). The 64x64 sliding window traverse in the selected section and collect all patches(Step 2 in Figure 3). These collected patches become batch and enter the detector. So, the output is not just one array, but arrays, number of which is same as the number of patches in batch. This method reduces the processing time per image from hours to a few seconds.

C. Re-train network with false-true samples

In Table 1, you can find the results of proposed method. It shows the great result when trained only 10 hours but it declines severely when trained 10 more hours. The reason is that the detector detects sea lions very well. You can see a sea lion with multiple brown spots in Figure 6. The detector regarded many parts of sea lion as independent sea lions. We have to re-train the network to think many edge parts of sea lion as background. So, we have to collect false-true samples from training images. The re-training framework is depicted in Figure 6 with explanation.

III. EXPERIMENT

A. Dataset and training

The dataset is NOAA sea lion dataset which was provided in KAGGLE sea lion counting competition [8]. Dataset consists of 948 air-view images(sizes of which are approximately 5600 x 3700) with pair of dotted and non-dotted. We separate the dataset into training (0-749) and test (750-947). Our network consists of CNN layers and fully connected layers. Each CNN layer has rectified linear unit for activation and 2x2max pooling layer for compression. For better performance, we embedded batch normalization before each rectified linear

TABLE I. COMPARISON

METHOD	*M.RMSE*	*TIME*
64x64 single image(10h training)	19.47	18.98
64x64 single image(20h training)	19.98	19.53
16x16 single image(10h training)	27.37	5.32
16x16 single image(20h training)	25.56	6.02
Proposed(10h training)	**19.02**	10.58
Proposed(20h training)	27.04	11.09
Proposed(10h additional training with false-true samples)	20.03	10.68

unit. After CNN layers, it follows fully connected layers with 0.5 drop-out probability. The final output is produced after softmax function. When training, output is evaluated by softmax cross-entropy function and optimized by adam-optimizer [7]. We set the adam learning rate as 1e-3. Before training, we extracted sea lion samples from dotted images. We applied blob detection to get position of each sea lion. Using this center position of each sea lion, sea lion samples are cropped with 80x80 size. We randomly sampled background patches for negative samples. Also patches near sea lions are sampled for hard-negative samples. We applied data augmentation in training. We used up-down flip and left-right flip with 50% probability. And modified brightness and contrast. Especially, when training classifier, we randomly cropped 64x64 size patches from 80x80 size initial samples. It improves the accuracy when detected position is not exactly located in center of sea lion. But for training detector, we cropped 64x64 initial training patch centered at 80x80 patch. This is for accurate detection of center of sea lion. One-hot vectors are used for loss-function of detector and classifier(length-2 for detector, length-4 for classifier).

B. Performance comparison

In this section we compare the accuracy and time-consumption in various aspects. We evaluated our method's performance in test dataset(750-947) by calculating M.RMSE and average time per image. M.RMSE is derived by following equation.

$$RMSE_i = \sqrt{\frac{\sum_c (n_i^c - \tilde{n}_i^c)^2}{N_c}} \qquad M.RMSE = \frac{\sum_i RMSE_i}{N_i}$$

c is class number(1-4) and i is image number(750-947). N is total number of class or image. n_i^c is predicted number of class c objects in image i. We compared ours to single patch based networks to prove the power of multi-viewpoints. For fair comparison, each method was trained equally 10-hours and 20-hours. First, we analyze the effect of training time. As we mentioned in section 3-C, result of our method severely got worsen through 10 more hours of training. 64x64 single patch method and 16x16 single patch method shows relatively consistent results regardless of training time. After re-training with false-true samples about 10 hours, our method showed

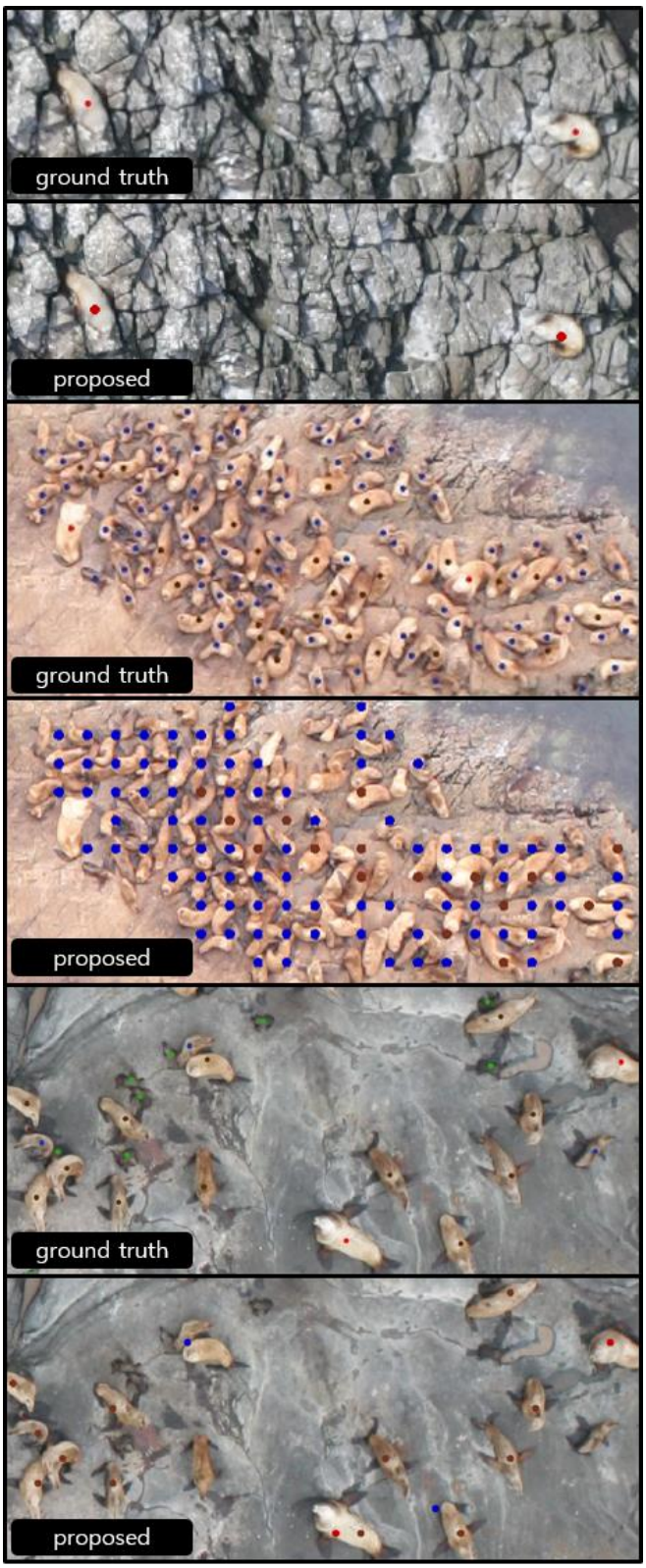

Fig. 7. Odd rows are ground truth and even rows are visualized results of proposed method. In the first result image, our method successfully detected two adult male sealions in confusing rocky background texture. And second and third result images also verify great accuracy of our method in complex scene compared to ground truth.

competitive result again. From now, the result of our method indicates the result after re-training. We compared ours with 64x64 single sliding window method. Although, it uses only one image patch as input, it has much higher resolution and context. Our proposed method used three 16x16 patches, which is 1/16 size of 64x64 patch. But as you can see in Table 1, our method showed competitive RMSE score to single 64x64 patch based method. And the time is almost twice faster. 16x16 single sliding window method is slightly faster than our method. But our accuracy outperforms it. Relatively lower speed is because of sequential cropping and resizing process. Parallel processing could accelerate the speed of our method. You can see the visualized detection and classification results of our method in Figure 7.

IV. CONCLUSION

In this paper, we introduced the framework to detect small animals in air-view image. There are three main contributions. First, inputs with different viewpoint and resolution improved speed without speed bottleneck. Second, fully utilized the ability of GPU by collecting patches as input batch. It significantly reduced processing time. And third, we devised own re-training method using training dataset by collecting false-true error patches. Our framework could be applied to other works, which require detecting and classifying small objects(e.g. face detection and age classification among many audiences). Our future work is implementing CNN structure that could consider wide context instead of small patches.

ACKNOWLEDGEMENT

This research is supported by Ministry of Culture, Sports and Tourism(MCST) and Korea Creative Content Agency(KOCCA) in the Culture Technology(CT) Research & Development Program 2017

REFERENCES

[1] K. Simonyan, A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition”, ICLR 2015.

[2] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, Andrew Rabinovich, “Going Deeper with Convolutions”, CVPR 2015.

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, “Deep Residual Learning for Image Recognition”, CVPR 2016.

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition, ECCV 2014.

[5] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”, NIPS 2015.

[6] Liu, Wei and Anguelov, Dragomir and Erhan, Dumitru and Szegedy, Christian and Reed, Scott and Fu, Cheng-Yang and Berg, Alexander C, “SSD : Single Shot MultiBox Detector”, ECCV 2016.

[7] Diederik P. Kingma, Jimmy Ba, “Adam: A Method for Stochastic Optimization”, ICLR 2015

[8] https://www.kaggle.com/c/noaa-fisheries-steller-sea-lion-population-count/data