Anti-UAV

The 1st Anti-UAV Workshop & Challenge

Zhao Xingjie
Sun Ting

Sigpro Lab
Xi'an Jiaotong University, XJTU

Outline

- Anti-UAV Challenge Overview
- Dataset Analysis
- UAV Detection from Video
- Experiments
- Visualization
- Future Work

Anti-UAV Challenge Overview

- Development Phase(100 sequences):

  Split into two datasets:

  80 sequences for training(70487 frames)

  20 sequences for validation(18522 frames)

- Final phase(60 sequences):

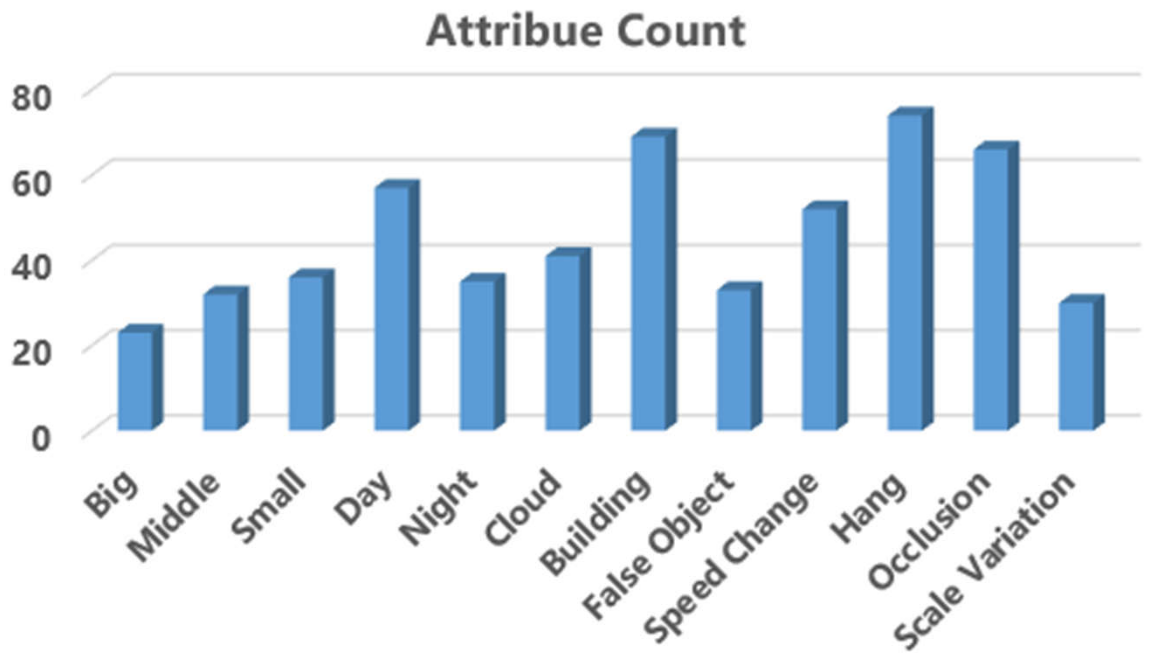
  Use previous trained model to detect UAV from test-challenge sequences

- Evaluation metric:

  Frame-wise: $$acc = \frac{1}{T}\sum_t (IoU_t * \delta(v_t > 0) + p_t(1 - \delta(v_t > 0)))$$

RGB videos vs IR videos

Dataset Analysis

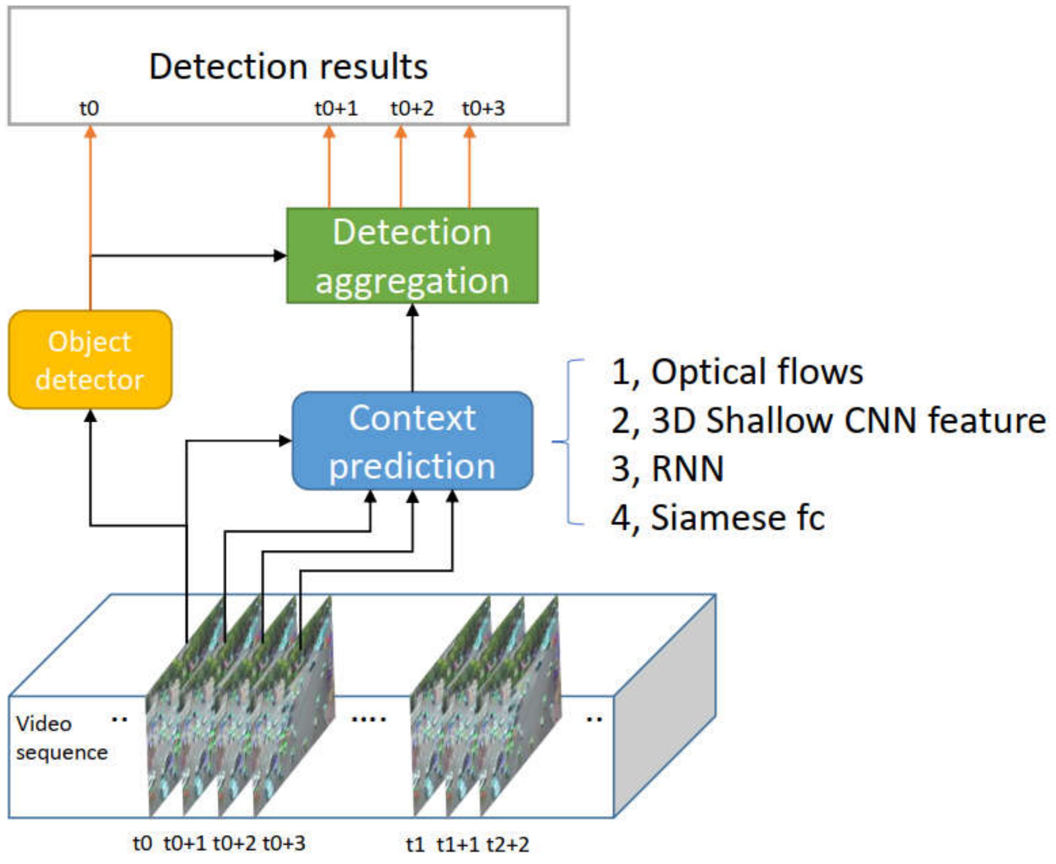


From the visualization and videos, we learn that:

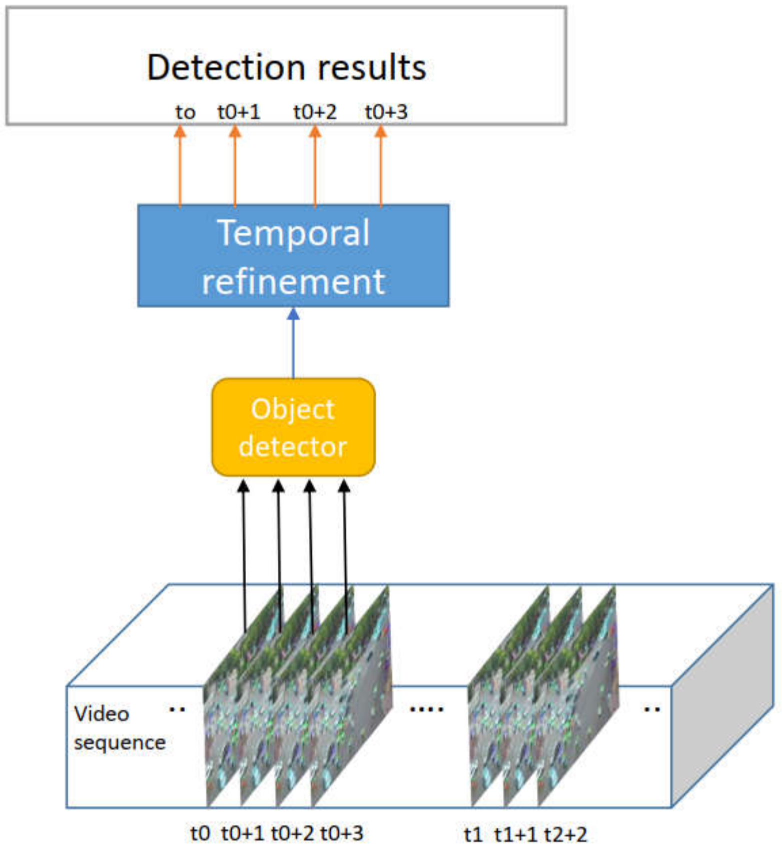
Different Challenges:

- Varying sizes
- Varying ratios
- Motion blur
- Fast motion
- Indistinguishable background

UAV Detection from Video



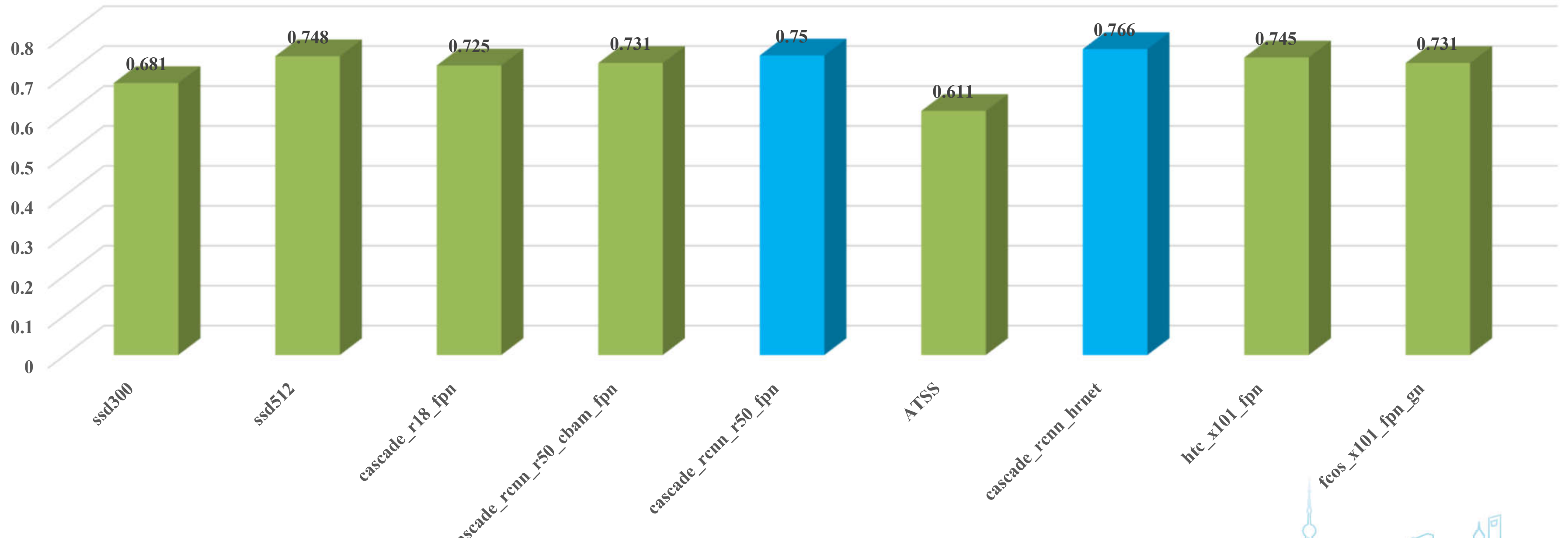
(1) clip-level detection



(1) frame-level detection

Frame-level detection:

Based on the MMDetection[1], we use different methods on 80 sequences training dataset, and evaluate on the 20 sequences validation dataset.



[1] Chen K, Wang J, Pang J, et al. MMDetection: Open mmlab detection toolbox and benchmark[J]. arXiv preprint arXiv:1906.07155, 2019.

Improve the Performance of Detector

□ OHEM（**Online Hard Example Mining**）：

The background and target in the dataset are highly similar, and there are hard samples (False Positive, negative examples are divided into positive examples). In the training process, according to the size of the loss of each sample, the relatively large loss is taken as the hard example, and these hard examples are trained again.

□ GN（**Group Normalization**）：

BN is normalized in the dimension of batch, but the batch size is generally different during training, validation, and testing, which leads to inconsistencies in these three stages. However, GN has nothing to do with batch size, which avoids this problem.

□ Model Ensemble：

- Ensemble different epochs of the same model
- Ensemble different models

OS	：	Ubuntu16.04 LTS
CUDA	：	10.0
CUDNN	：	7.5
PyTorch	：	1.0
GPU	：	NVIDIA GTX1080Ti（11GB Memory）
Dataset	：	80 sequences for training, 20 sequences for validation

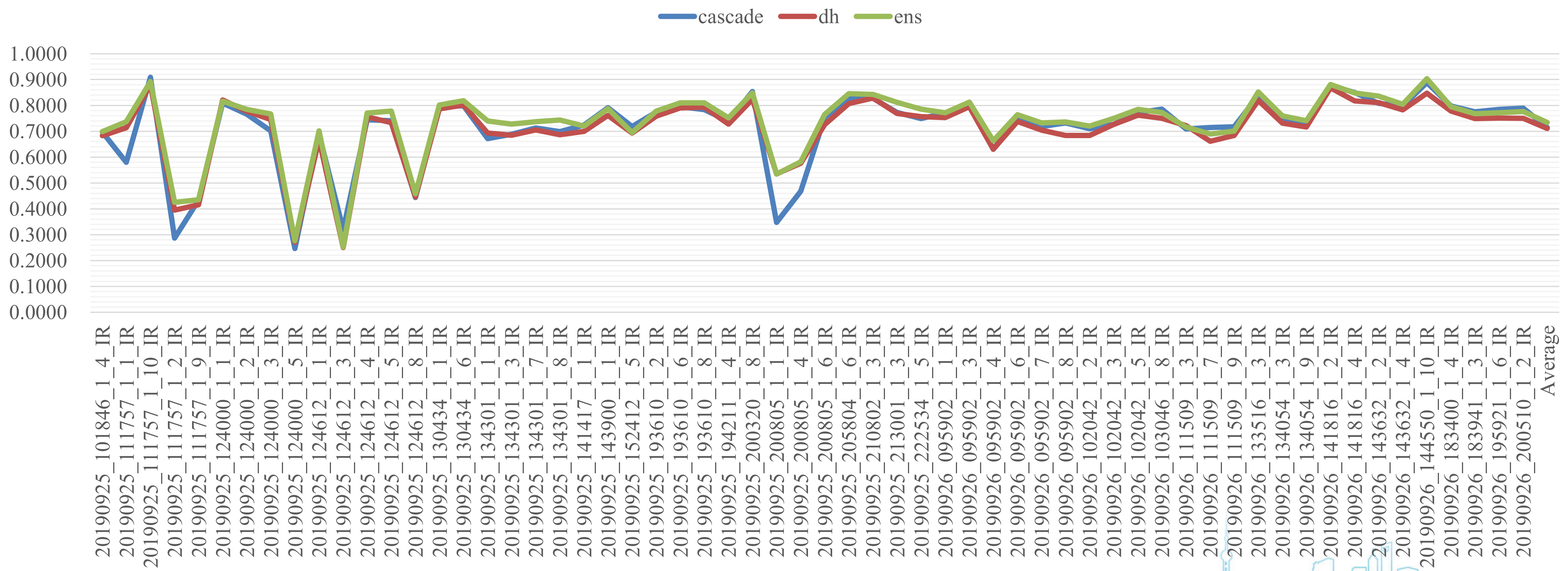
Method	**acc（%）**	**delta（%）**
Cascade_hrnet_fpn(baseline)	**76.6**	——
+Focal Loss	**76.5**	**-0.1**
+OHEM	**77.8**	**+1.2**
+GN	**77.3**	**+0.7**
+OHEM+GN	**79.9**	**+3.3**

Experiments

In the Final phase， we test different models from Development Phase. We also ensemble DH_r50_fpn and Cascade_hrnet_fpn+OHEM+GN.

Method	**acc（%）**
HTC_x101_fpn	**69.36**
DH_r50_fpn	**71.13**
Cascade_hrnet_fpn+OHEM+GN	**71.50**
Model_Ensemble	**73.46**

AP curve of different methods:



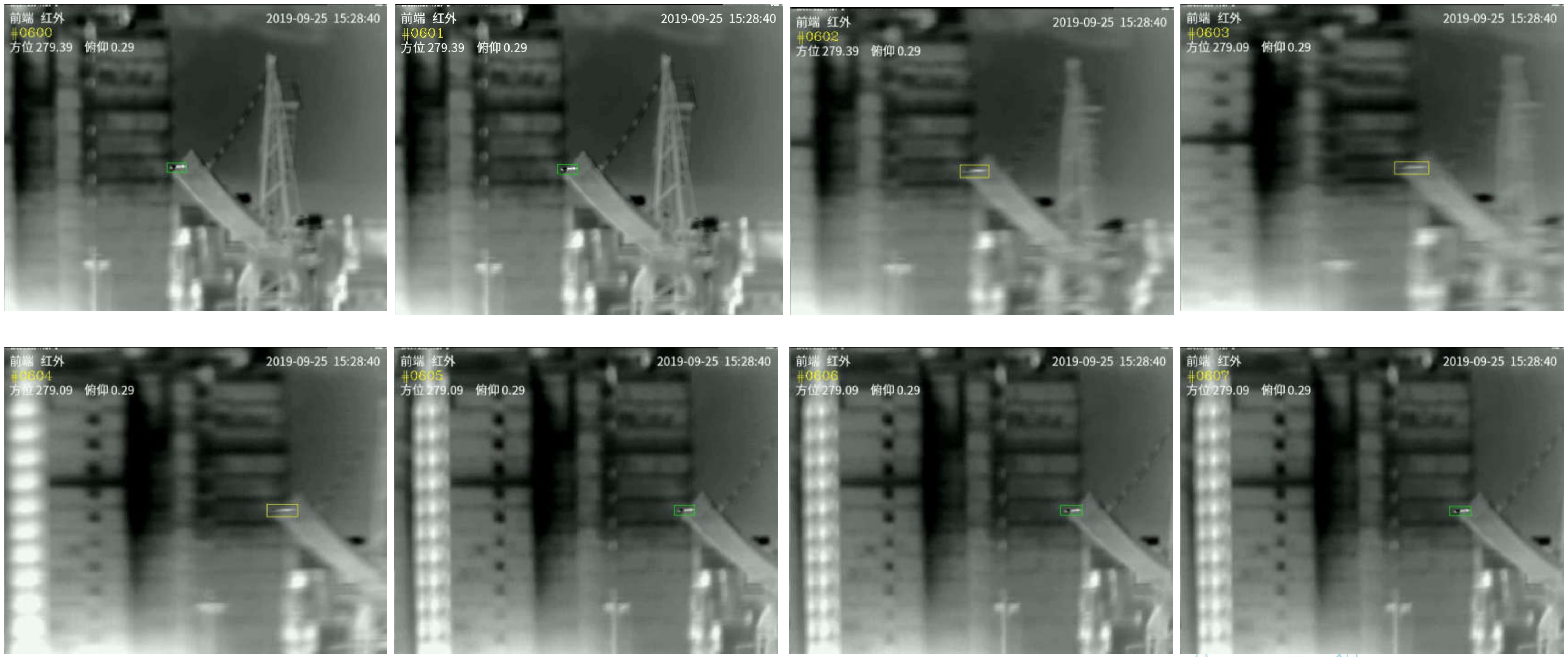
Future work

We will keep focusing on enhancing detection for small objects in infra-red videos in the future.

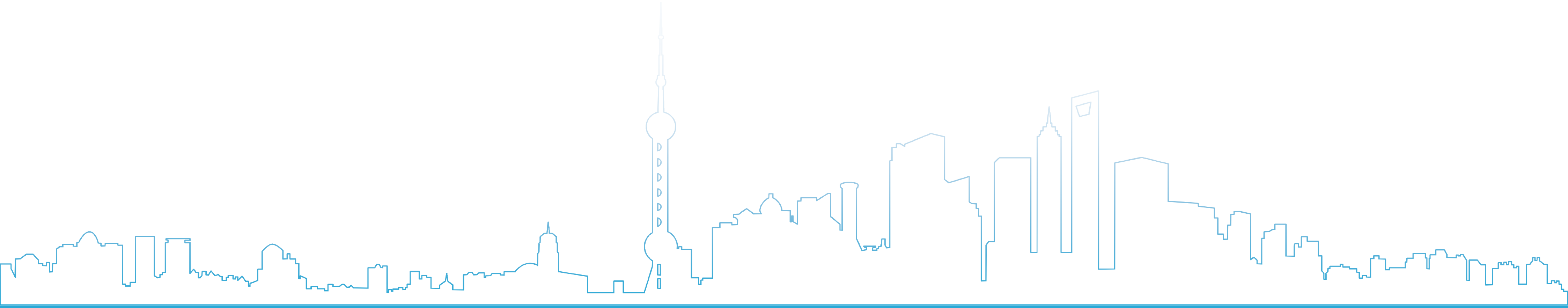
The target detection in video has real-time requirements. We will try to use Single-Stage algorithm to achieve this goal.

Visualization

Green for Cascade R-CNN， yellow for Double Heads R-CNN



前端 红外 2019-09-25 15:28:40 #0600 方位 279.39 俯仰 0.29
前端 红外 2019-09-25 15:28:40 #0601 方位 279.39 俯仰 0.29
前端 红外 2019-09-25 15:28:40 #0602 方位 279.39 俯仰 0.29
前端 红外 2019-09-25 15:28:40 #0603 方位 279.09 俯仰 0.29
前端 红外 2019-09-25 15:28:40 #0604 方位 279.09 俯仰 0.29
前端 红外 2019-09-25 15:28:40 #0605 方位 279.09 俯仰 0.29
前端 红外 2019-09-25 15:28:40 #0606 方位 279.09 俯仰 0.29
前端 红外 2019-09-25 15:28:40 #0607 方位 279.09 俯仰 0.29



Thanks!