

# Robotic Interestingness via Human-Informed Few-Shot Object Detection

Seungchan Kim, Chen Wang, Bowen Li, Sebastian Scherer

**Abstract**— Interestingness recognition is crucial for decision making in autonomous exploration for mobile robots. Previous methods proposed an unsupervised online learning approach that can adapt to environments and detect interesting scenes quickly, but lack the ability to adapt to human-informed interesting objects. To solve this problem, we introduce a human-interactive framework, AirInteraction, that can detect human-informed objects via few-shot online learning. To reduce the communication bandwidth, we first apply an online unsupervised learning algorithm on the unmanned vehicle for interestingness recognition and then only send the potential interesting scenes to a base-station for human inspection. The human operator is able to draw and provide bounding box annotations for particular interesting objects, which are sent back to the robot to detect similar objects via few-shot learning. Only using few human-labeled examples, the robot can learn novel interesting object categories during the mission and detect interesting scenes that contain the objects. We evaluate our method on various interesting scene recognition datasets. To the best of our knowledge, it is the first human-informed few-shot object detection framework for autonomous exploration.

## I. INTRODUCTION

Interestingness recognition is a crucial component of autonomous mobile robot exploration. Robots should perceive the surroundings by processing visual information quickly and estimate the interestingness of the scenes to make decisions for next actions and plans. Take a robot that explores underground tunnel as an example (first row of Fig. 1). While an uninteresting scene like Fig. 1 (a) may not change the course of the robot actions, interesting scenes that contain the wooden door as in Fig. 1 (b) or a dark entrance hole as in Fig. 1 (c) would affect the next actions of the robots, such as moving towards the door or hole.

There have been various approaches to modeling visual interestingness using the concepts of saliency [1], [2], memorability [3], and novelty detection [4]. However, these previous algorithms were not apt for robotic applications, because they were unable to adapt to unknown environments quickly and perform real-time responses to the rapidly changing scenes. Recently, Wang *et al.* proposed an unsupervised learning approach that detects visually interesting scenes in an *online* manner [5], [6]. The core of this algorithm is the use of a visual memory module that adapts to the surrounding environment quickly and remembers input scenes efficiently, so that the robot can recall the past and detect novel, unusual scenes as interesting in real-time [6]. Another notable feature is that, even if the robot once detects a specific scene as

The authors are with the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213, USA. seungch2@andrew.cmu.edu; chenwang@dr.com; bowenli1024@gmail.com; basti@andrew.cmu.edu



Fig. 1: Exemplary images from robots' exploration in the DARPA Subterranean Challenge. Some scenes, e.g., (a) are uninteresting, while there are interesting scenes e.g., (b), (c), (d), (e), and (f).

interesting, it loses interest over following similar scenes gradually if they appear repetitively, just like humans lose interest when exposed with a series of similar scenes.

However, one limitation of this algorithm is that it does not have the capability to adapt to human-informed interesting objects. Take this scenario as an example: an unmanned aerial vehicle (UAV) is exploring an underground tunnel and encounters scenes that look interesting and contain different objects as shown in Fig. 1 (d), (e), and (f). The previous unsupervised online learning algorithm [5] enables the UAV to detect these scenes as interesting, but it can't specify which objects within the scenes make them actually interesting. Depending on the missions and human operator's interests, the targeted interesting objects might differ with respect to the context. To solve this problem, we aim to take human operators' feedback into account for better recognition of interesting scenes and objects.

The challenge is that, before a mission starts, *both* the robot and human operator may not have information about what the interesting objects would be. During the exploration, the robot may discover a completely novel object, which is possibly uncommon for existing large scale training set [7] and even could belong to an unknown category to the human operator. However, it could still be interesting to the operator, and there is a need for the robot to learn about the object quickly *during the mission* and detect similar objects in the later phase of the exploration. Hence, traditional object detection approaches [8], [9], [10] trained for fixed and predefined categories on existing large scale datasets [7] are inapplicable to this situation.

We suppose that human's few online inputs on the object-

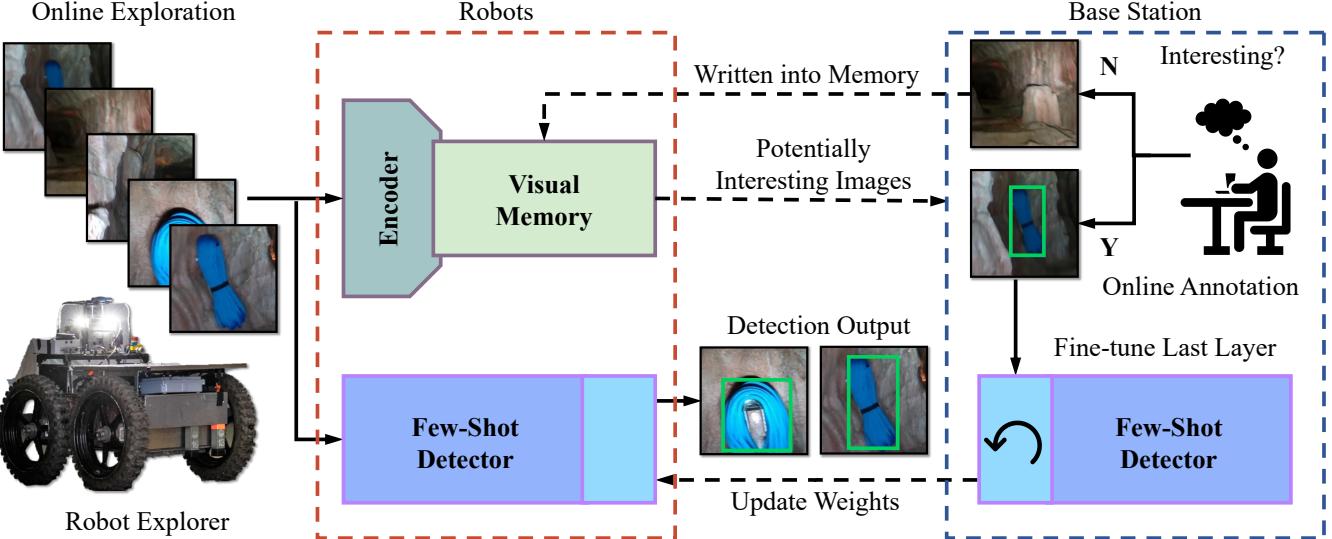


Fig. 2: Diagram of AirInteraction. The robot explorer perceives continuous streams of images, which first go through the unsupervised visual interestingness module. The interesting scenes detected by the visual memory are sent to the base station, where the human annotator can decide if the image is interesting or not. Uninteresting images are sent back to the robot and written into the memory, which prevents the false judgement of similar scenes afterwards. The interesting objects in the real interesting images are annotated by the human (drawing tool is given). Then the last layer of the few-shot object detector is fine-tuned on the novel object using the provided annotations. The updated weights of the final layer are copied to the classifier onboard periodically, so that the robot can detect novel interesting objects in real-time during the mission.

specificity, (e.g., certain objects or specific image parts that make the scene interesting), can be of great assistance to robot exploration. Specifically, when a robot detects an interesting scene (image) online, it sends the potential interesting image to a human operator at the base station. When the human operator thinks an image is interesting, the operator is prompted to draw a bounding box around the interesting object/part within the image. This image and bounding box annotation are fed into a few-shot object detector [11], which can quickly learn an object detection model only using a very few labeled examples. Since human annotations can be expensive, we restrict the model to learn novel, unseen objects using just one, two, or three examples per category.

In summary, our contributions are:

- To solve the problem of existing algorithm that can only detect predefined object categories, we propose a human-interactive framework, AirInteraction, for autonomous robots to detect visually interesting and human-informed objects during online exploration.
- To reduce human efforts and communication bandwidth, we design three components: (1) an online unsupervised algorithm running on the exploration robot to detect visually interesting scenes, (2) a human-robot interaction interface running on a base-station to process operator’s feedback for interesting objects, and (3) a few-shot object detector running across the robot and the base-station to enable fast training of human-informed objects during the robotic mission.
- We test the effectiveness of AirInteraction on various challenging datasets and perform ablation studies to verify the necessity of each component.

In Section II, we review prior works on robotic interestingness and few-shot object detection. Our new method, human-in-the-loop online visual interestingness with few-shot object detection is presented in Section III. The experimental setup, results of our method and baselines, and the ablation study are shown in Section IV, followed by the limitation of our framework and possible solutions in Section V.

## II. RELATED WORK

### A. Visual Interestingness

Although prior works [12][13] have called the extent to which an image/video captivates or holds one’s attention as visual interestingness, the strict definition and quantification of visual interestingness hasn’t been globally agreed yet. Since the notion of interestingness is inherently subjective and non-trivial [13], there have been many different approaches to model visual interestingness using various concepts. Isola *et al.* [3] and Bylinskii *et al.* [14] focused on the concept of memorability, investigating the intrinsic and extrinsic factors that make images remembered by humans’ memory. Borji *et al.* [1] and Zhao *et al.* [2] focused on the concept of saliency, investigating features of subparts, objects, or regions within an image that stand out from neighboring parts, and reviewed saliency-based visual attention that drives the shift and focus of human gaze. Gygli *et al.* [15] proposed computational evidence for different visual cues that constitute interestingness of images, especially focusing on aesthetics, unusualness, and preferences. Hsieh *et al.* [16] examined the interaction between visual and social interestingness, focusing on virality, popularity, and aesthetics features. However, these aforementioned works

have focused on *static* properties of images, which is not suitable for situations that require adaptation to the changing distributions. Previous works [5], [6] proposed a new concept of visual interestingness for robot exploration with online adaptation to changing distributions. It can report the potentially interesting frames, but fail to detect user-informed objects in real-time. In this paper, we will solve this problem by incorporating a few-shot object detection model.

### B. Few-Shot Object Detection

Object detection aims to estimate the locations and pre-defined categories of objects in an image. With the emergence of deep learning-based methods, there have been rapid progress of object detection algorithms, mainly divided into two branches of algorithms: (1) two-stage approaches [17], [8], [18], [19] that separate region proposal generation and proposal classification/regression, (2) one-stage approaches [9], [10] that design an end-to-end neural network to tackle the problem. However, the limitation of the general object detection is the requirement of the large scale annotated training data. Moreover, the object categories are fixed and predefined after training.

Along with the recent progress in few-shot learning [20], [21] that learns to generalize with only a few novel data, there have emerged various approaches to tackling few-shot object detection task. A few-shot detector aims to detect novel object classes using merely a few labeled examples. One paradigm for few-shot object detection is transfer learning [11], [22], [23], which pretrains a base model on base classes using abundant data, and adapts the model via fine-tuning on few labeled examples. Another paradigm is meta-learning [24], [25], [26], [27], [28], which aims to extract meta-level knowledge during base training that enables efficient learning on novel classes with a few examples.

## III. METHODOLOGY

### A. Overview

As shown in Fig. 2, AirInteraction is composed of three components, including (1) an online unsupervised learning algorithm running on the robot that detects interesting scenes during exploration in real-time, (2) a human-robot interactive interface at the base station, and (3) a few-shot object detector that learns human-informed objects online.

The workflow of AirInteraction is also simple but efficient. Due to the uncertainty of working environments, we cannot guarantee the network condition, thus we reduce the communication bandwidth requirement by putting most of the computation on the robot and only send necessary information to the base-station. Specifically, the robot detects potentially interesting scenes by employing the online interestingness detection algorithm and only sends the images with highest probability to be interesting to the base-station.

If the network is interrupted, the interesting scenes will be stored in a stack and cover those images with lowest probability. The images with higher probability will be sent in higher priority once the network is established again. If the human operator with the base-station thinks an image is

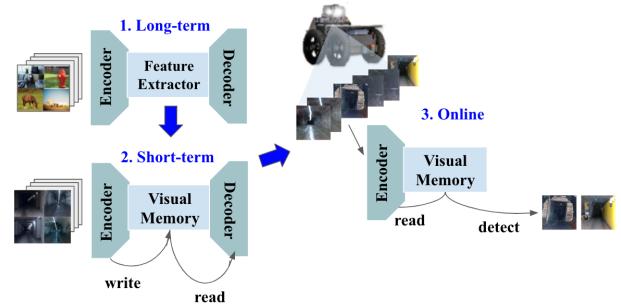


Fig. 3: The structure of the unsupervised online learning model. It detects the potentially interesting scenes based on a visual memory module. The images that are not well recalled by the online updated memory are taken as interesting, which will be sent to base-station.

interesting, the operator will be prompted to select the region of interests, such as a novel object. The selected regions (bounding boxes) and the object labels are then taken as training samples to fine-tune the few-shot object detector.

The few-shot detector is fine-tuned on the base-station and a mirror copy model runs on the robot. The human operator's annotations are dynamically added to the training samples' pool to update the model parameters, which are sent to the robot periodically to detect similar objects. We next explain more details about the three modules, respectively.

### B. Online Unsupervised Interestingness Detection

To reduce the communication requirement, we leverage the unsupervised online learning model [5], [6] to select potentially interesting frames. This approach is composed of a three-stage learning architecture, i.e., (1) the *long-term* learning, which acquires general world-level knowledge on large scale training data, (2) the *short-term* learning, that learns prior task-dependent knowledge via a visual memory module from hundreds of uninteresting images in several minutes, and (3) the online learning, which processes image stream and updates the memory module in real-time to adapt to the environments.

The key of this architecture is the visual memory, which saves the knowledge about the uninteresting scenes during the short-term learning, and outputs lower confidence when novel scene appears suddenly during the online mission execution. One important feature of this system is a loss of interests over repetitive scenes, i.e., when a robot sees an interesting image, it is quickly written into the memory, so that the interestingness level of similar following scenes decrease gradually. However, the interestingness recognition system [5], [6] has no interaction with the operators and it cannot detect specific human-informed objects.

When the exploration robot perceives the surrounding environment, it processes continuous streams of image. Each image vector  $\mathbf{x}(t) \in \mathbb{R}^{C \times W \times H}$  goes through the visual memory module  $\mathbf{M} \in \mathbb{R}^{N \times C \times W \times H}$  of the interestingness algorithm, where  $C, W, H$  are number of channels, width, and height of the image, and  $N$  is the number of cubes in the visual memory. When the robot sees a new scene  $\mathbf{x}(t)$  at

time  $t$ , the visual memory module  $\mathbf{M}$  is updated as follows:

$$\mathbf{M}_i(t) = (1 - w_i) \cdot \mathbf{M}_i(t-1) + w_i \cdot \mathbf{x}(t), \quad (1)$$

where  $w_i$  is the  $i$ -th element of the vector  $\mathbf{w} \in \mathbb{R}^N$

$$\mathbf{w} = \sigma\left(\gamma_w \cdot \tan\left(\frac{\pi}{2} \cdot \mathcal{D}(\mathbf{x}(t), \mathbf{M}(t-1))\right)\right), \quad (2)$$

where  $\sigma$  is the softmax function,  $\gamma_w$  is a learning speed control parameter, and  $\mathcal{D}$  is the cosine similarity function. This *writing* process [5], [6] enables the visual memory module to incrementally learn new scenes without forgetting old scenes.

Concurrently, when the robot sees a scene  $\mathbf{x}(t)$ , it needs to recall the memory and compare the new scene with the old scenes within the memory. Within this memory *reading* process [5], [6], the reading score is defined as

$$f(t) = \sum_{i=1}^N v_i \cdot \mathbf{M}_i(t), \quad (3)$$

where  $v_i$  is the  $i$ -th element of the vector  $\mathbf{v} \in \mathbb{R}^N$

$$\mathbf{v} = \sigma\left(\gamma_v \cdot \tan\left(\frac{\pi}{2} \cdot \mathcal{S}(\mathbf{x}(t), \mathbf{M}(t))\right)\right), \quad (4)$$

where  $\gamma_v$  is also a learning speed control parameter for reading, and  $\mathcal{S}$  is maximum cosine similarity operator between  $\mathbf{x}(t)$  and all possible translations of  $\mathbf{M}(t)$ , which is calculated from element-wise multiplication of their Fourier transforms.

Intuitively, when the memory reading confidence score is low, the scene is more likely to be interesting since the scene has never been visited before, and vice versa. In practice, we take the cosine similarity between  $f(t)$  and  $x(t)$  as the reading confidence. All the scenes that were previously seen are written into the memory; this prevents robots over-detecting similar scenes as interesting. Even if a scene is interesting at first glance, robot still loses interests over the scene, if the scene is repetitively visited.

We choose the visual memory learning scheme [5], [6] as the first component of AirInteraction, not only due to the high efficiency of the algorithm, which enables real-time online processing of image streams, but also because it can make prior judgments so that only a few potentially interesting scenes will be sent to the base station. Using the visual memory, the robot can respond to novel scenes only, ignore the repetitive or similar scenes, and avoid the redundant request for human feedback.

### C. Human-Robot Interactive Interface

We design the interface for human-robot interaction between the robot explorer and human operator based on OpenCV [29] and ROS [30]. A screenshot of the visualization and annotation tool is shown in the Fig. 4. When the interesting scenes detected by the robot are sent to the base station, the scenes are queued and being shown to the human operator one by one. If the human operator presses the key  $N$ , which denotes *not interesting*, the image is sent back to the robot and written into the memory, which prevents similar following scenes from being detected as interesting by the robot.

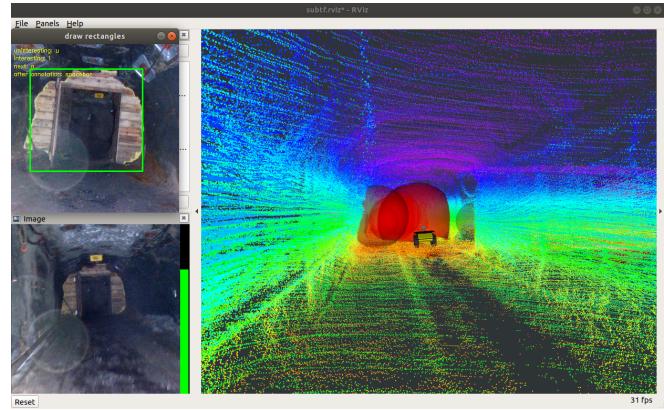


Fig. 4: Human-robot interface for annotation and surveillance.

If the human operator presses the key  $I$ , which means the human operator thinks the scene is *interesting*, the pop-up screen of the image appears where the operator can draw bounding box of the interesting object in the image. The annotation information of the image and the location of the bounding box are sent to the few-shot object detector running on the base station, which will be explained next subsection.

Our interface is designed to make the human operator's interaction with robot as easy and simple as possible. Using this visualization tool, any inexperienced human operator can give feedback to the robot with minimal efforts and actions by just clicking the image screen and using a mouse to draw the bounding box.

### D. Few-shot Object Detector

The human-informed object annotations and the image pairs serve as few labeled example inputs for few-shot object detector, which is equipped in the base station. We further explain the details about the choice of the few-shot object detection model and techniques to improve its performance.

*1) Two-stage Fine-tuning Approach:* In this work, we employ one of the transfer-learning based few-shot object detection model, two-stage fine-tuning approach (TFA) [11] as a baseline, which is known to have state-of-the-art performance on benchmarks. The two-stage refers to (1) the base training stage which jointly trains feature extractor and box predictor for base classes, and (2) the fine-tuning stage which trains box predictor for novel classes. For the base model, we follow the same architectures proposed by [11], where Faster R-CNN [8] is adopted as a base detector with Resnet-101 [31] backbone.

During the fine-tuning stage, the feature extractor of the model is fixed, and only the final layer of the box predictor is fine-tuned. While training for base classes requires abundant data, training for novel classes via fine-tuning the final layer only requires using few labeled examples. We use the human-informed image-annotation as the labeled examples and feed them into the trainer to fine-tune the final layer of the model.

*2) Weighted Mixture of Batch for Fine-tuning:* When fine-tuning the last layer of the model, previous work [11] used the balanced mixture of base class images and novel

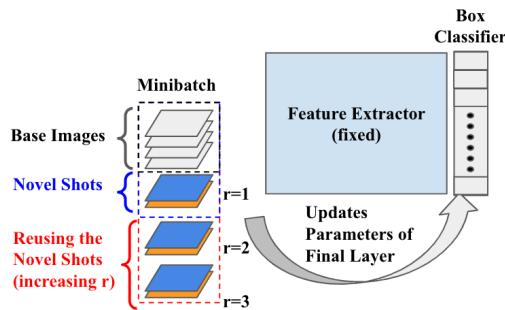


Fig. 5: The combination of base images and novel image shots compose the batch for the fine-tuning training. We parameterize the frequency of using novel image shots compared to base images, and use the novel shots up to 3 times more frequently than base images.

class images as a minibatch for training. However, our experiments show that this often cannot produce satisfactory results for novel classes. Instead, AirInteraction employs a weighted mixture of batch for fine-tuning. Let's denote the number of base classes and novel classes as  $B$  and  $N$ , and we use  $K$  instances per category during fine-tuning. The random mixture of  $BK$  base images and  $NK$  novel images are used. If the minibatch size is  $m$ , each minibatch is a random combination of base and novel images, and there are  $C_{(B+N)K}^m$  combinations. In this work, we use a variant of this scheme by increasing the frequency of using novel (human-informed) images when generating a minibatch, rather than using random uniform mixture. Let's denote the ratio of novel images compared to base images as  $r$  ( $r \geq 1$ ). Then there can be  $C_{(B+rN)K}^m$  combinations of minibatches to be used for fine-tuning. Note that we are not changing the number of instances per novel category, but just simply reusing the existing instances more frequently to generate minibatches.

3) *Parameter synchronization between robot and base station:* We have two identical few-shot object detectors in the system, one in the base station, and the other in the robot. Since training the object detection model is computationally heavy for the robot, we perform the fine-tuning process in the base station. Note that the parameters of the pretrained features are fixed, both in the robot and the base station, and only the parameters of the final layer are updated. Therefore, it suffices to synchronize the final layer parameters of the object detector in the base station and the one in the robot. The dimension of the final layer is comparatively small, and can be sent easily to the robot for synchronization.

#### IV. EXPERIMENTS & RESULTS

##### A. Dataset

The experiments are conducted on four authoritative challenging datasets, as shown in Fig. 6, to verify the effectiveness of our system.

1) *SubT Tunnel and SubT Urban Datasets:* The two datasets are recorded during the DARPA Subterranean (SubT) Challenge, which requires competitors to build autonomous robotic systems to search and explore the subterranean environments like tunnel and urban underground.



Fig. 6: Representative images from each dataset. From 1st to 4th row: SubT Tunnel, SubT Urban, Scott Reef 25, Drone Filming

In this challenge, the autonomous robots are prompted to search for different objects. We use the datasets recorded by two autonomous UGVs from the Team Explorer (Carnegie Mellon University and Oregon State University). For SubT Tunnel circuit, we chose door, hole, surveillance camera, and fire extinguisher as interesting objects, and for SubT Urban dataset, we chose helmet, backpack, rope, drill, vent, and human survivor as interesting objects.

2) *Scott Reef 25 Dataset:* The Scott Reef 25 dataset is a sequence of top-down view images recorded by autonomous underwater vehicle (AUV) at Scott Reef in Western Australia. There are five labeled objects in this dataset, in which we take sand as uninteresting and the other objects including finger coral, mushroom coral, weed, and debris as interesting.

3) *Drone Filming Dataset:* It is an outdoor dataset recorded by quadcopters that follow moving targets during real-time aerial filming. In this work, we select three different object categories (van, truck, bicycler) as interesting.

##### B. Baseline, Metrics, and Hardware

Since AirInteraction is the first human-informed robotic interestingness detection system, we lack off-the-shelf systems to compare. Therefore, we will take the few-shot object detection method TFA [11] as a baseline. For fairness, we adopted the default parameters for the TFA. To test the performance of few-shot object detector trained on the inputs from the human operator, we use the traditional evaluation metric, Average Precision (AP) score [7] for each class of objects and across the classes. For each dataset, we measure and average the performances over the object classes. For fine-tuning, we used 1 Nvidia GeForce RTX 2060 GPU.

##### C. Performance with Fixed Number of Shots

Since the online annotation is costly, we assume that human operator can give annotation feedback for up to 3 scenes per class. Therefore, we take very challenging settings to demonstrate the effectiveness of our method, i.e., only  $k = 1, 2, 3$  shots are provided in the experiment. In addition, we also report the performance using the weighted mixture

TABLE I: Performance of AirInteraction and TFA on all datasets. Bold values denote the best performance in each number of shot.

Dataset	Model	1 Shot	2 Shots	3 Shots
SubT Tunnel	Ours ( $r = 1$ )	5.21	10.55	13.65
	Ours ( $r = 2$ )	12.39	14.78	14.29
	Ours ( $r = 3$ )	<b>13.38</b>	<b>17.56</b>	<b>17.81</b>
	TFA [11]	7.52	13.85	17.20
SubT Urban	Ours ( $r = 1$ )	4.35	8.27	9.52
	Ours ( $r = 2$ )	8.32	10.20	11.51
	Ours ( $r = 3$ )	<b>12.53</b>	<b>13.68</b>	<b>14.95</b>
	TFA [11]	8.54	10.56	10.82
Scott Reef 25	Ours ( $r = 1$ )	8.37	10.02	34.72
	Ours ( $r = 2$ )	<b>35.28</b>	31.77	<b>42.49</b>
	Ours ( $r = 3$ )	32.48	<b>34.62</b>	37.74
	TFA [11]	28.25	30.92	36.97
Drone Filming	Ours ( $r = 1$ )	12.60	63.52	51.82
	Ours ( $r = 2$ )	40.03	83.41	83.25
	Ours ( $r = 3$ )	<b>62.68</b>	<b>89.36</b>	<b>84.38</b>
	TFA [11]	35.26	70.59	80.95

of batch for fine-tuning as described in Section III-D.2, i.e., the novel-to-base image ratio  $r = 1, 2, 3$  are used.

In the Table I, we reported the overall performance of AirInteraction, which is fine-tuned for 5 minutes to meet the quick deployment requirements. The results show that AirInteraction with  $r = 2$  and 3 achieve better performances than the TFA baseline in all three datasets, while AirInteraction with  $r = 1$  achieves lower performance than TFA.

In average, the AirInteraction with  $r = 2$  and  $r = 3$  show **10.8%** and **29.7%** higher performance than the TFA baseline across all datasets. AirInteraction with  $r = 1$  shows 34.4% lower performance than the TFA on average. This highlights the effect of tuning the ratio  $r$  when employing AirInteraction; reusing the novel shots provided by the human operator in the minibatches facilitates the improved performance to the original model. We also note that the performance of AirInteraction with  $r = 1$  is relatively lower than the TFA baseline, because the dynamic addition of human-informed shots during the training is not as stable as starting the training with novel shots in the first place. Increasing the ratio  $r$  can be thought of as a method to make up this shortcoming by using novel shots more frequently for training.

In Table II, we additionally reported the best performance of the AirInteraction model by object classes in the dataset. We observe that the AirInteraction is achieving comparatively good performance in detecting objects in Drone Filming and Scott Reef 25 datasets than SubT tunnel and SubT Urban datasets.

#### D. Performance with a Fixed Training Time

In Section IV-C, we report the model performance with fixed fine-tuning time. However, during autonomous exploration, the robots should be able to learn the novel object categories within a limited amount of time for quick deployment. In this section, we further report the model performance in Fig. 7 with a fine-tuning time of 1, 2, 3, 4, and 5 minutes. This means that the few-shot object detector should learn novel object categories within five minutes for

TABLE II: Best Object detection performance of AirInteraction for each class of object in each dataset.

Dataset	Class	Performance
SubT Tunnel	Door	19.28
	Hole	10.25
	Cctv	14.55
	Fire Extinguisher	18.50
SubT Urban	Backpack	14.38
	Helmet	18.25
	Rope	9.44
	Drill	11.69
	Vent	16.28
Scott Reef 25	Survivor	13.74
	Weed	32.54
	Finger Coral	53.15
Drone Filming	Mushroom Coral	43.16
	Van	98.52
	Truck	83.41
	Bicyler	88.11

exploration, which is acceptable for most tasks. In general, the performance of the object detector improved over time. In the case of  $r = 2$  and 3, the performance of the object detector started to saturate starting 3 minutes, while in the case of  $r = 1$ , the performance of object detector was low until 5 minutes. On average, across the datasets, the model performance of AirInteraction with  $r = 2$  at time  $t = 3$  and 4 minutes was 80.8% and 97.5% of its performance at  $t = 5$  minutes, respectively, and the performance of AirInteraction  $r = 3$  at time  $t = 3$  and 4 minutes was 85.6% and 94.8% of its performance at  $t = 5$  minutes.

#### E. Online Responsivity

Real-time performance is crucial for robotic systems, thus we next measure the online responsivity of the interestingness detection system. Specifically, we employ the metric Area Under Curve of Online Precision (AUC-OP) [5], which jointly considers online response, precision, and recall rate. The baseline is the online visual interestingness algorithm [5] without the object detector component, which detects the binary interestingness of the scenes only using a visual memory module. Basically, all the scenes that contain interesting objects are taken as interesting, and the others are not interesting.

In the Table III, we reported the AUC-OP scores of the baseline visual interestingness algorithm and AirInteraction in the different datasets. In the SubT Tunnel, SubT Urban, Scott Reef 25, and Drone Filming datasets, the AirInteraction showed better online response scores than the baseline. Note that we often accept several false positives to improve the recall rate and the metric of AUC-OP provides a variable  $\delta$  ( $\delta \geq 1$ ) to reflect this. Basically, higher  $\delta$  means more false positives are allowed and 1 means no false positive is allowed ( $\delta = 2$  is recommended for most applications).

#### F. Reducing Communication Bandwidth

The purpose of online unsupervised learning module is to select a few interesting scenes and send those scenes to

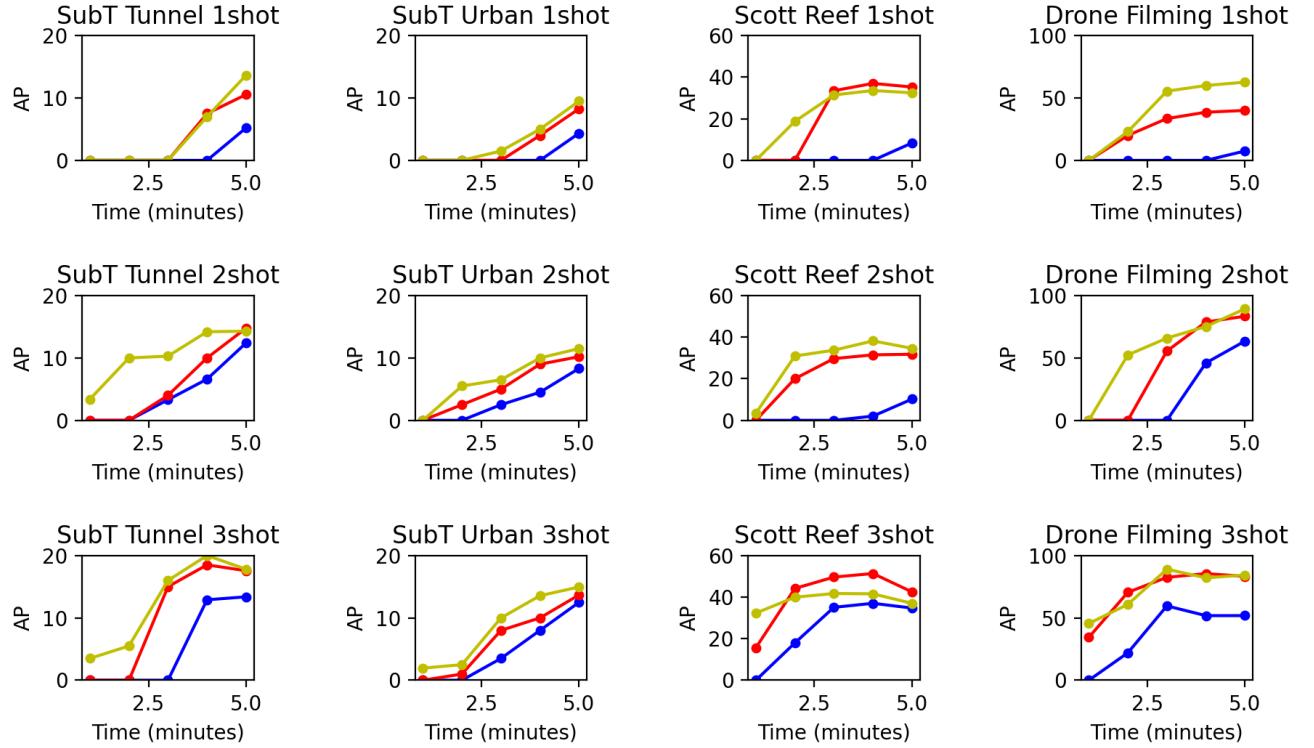


Fig. 7: Performance of few-shot object detector within five minutes. Blue, red, yellow graph each denotes AirInteraction  $r = 1, 2, 3$

TABLE III: The performance on AUC-OP score.

Dataset	Method	$\delta = 1$	$\delta = 2$	$\delta = 4$
SubT Tunnel	Baseline [5]	0.32	0.51	0.72
	Ours	0.36	0.53	0.81
SubT Urban	Baseline [5]	0.29	0.45	0.66
	Ours	0.35	0.44	0.69
Scott Reef 25	Baseline [5]	0.44	0.71	0.89
	Ours	0.58	0.75	0.85
Drone Filming	Baseline [5]	0.56	0.68	0.74
	Ours	0.59	0.70	0.88

the base station, thereby reducing communication bandwidth between the robot system and the human operator. In this section, we measure the reduction of communication bandwidth by comparing the ratio of the number of selected interesting scenes sent to the base station and the number of total scenes. As can be seen in Table IV, our system reduce the communication bandwidth 80 % up to 91% across datasets, which indicates its effectiveness.

TABLE IV: Percentage of images being sent

	SubT-Tunnel	SubT-Urban	SR25	DF
Ratio	0.15	0.11	0.20	0.09

## V. LIMITATION

One of the limitations of AirInteraction comes from the fact that it is using a fine-tuning based few-shot object detector. Under our specific choice of fine-tuning-based object detection model [11], the maximum number of novel object classes should be fixed; we set the capacity of new object categories as 10, but if the robot faces more objects during the mission, the capacity should be enlarged. Moreover, fine-tuning the model requires some amount of time to reach a good performance; although we limited the training time to five minutes, and showed that AirInteraction achieves good performance within this time constraint, it is better to shorten the amount of time learning a new category. To this end, deploying a few-shot object detection model without fine-tuning should be considered. Recent works on dynamic few-shot object detection without fine-tuning [28] or incremental meta-learning based few-shot object detection [26] can be our future alternatives.

## VI. CONCLUSION

We proposed the human-robot interactive framework, AirInteraction, that enables interesting scene recognition via human-informed few-shot object detection. Composed of three components, (1) online unsupervised learning module, (2) human-robot interaction interface, and (3) few-shot object detector, AirInteraction aids mobile unmanned exploration robots to learn novel interesting objects during the mission only relying on minimum amount of human feedback. Our

additional techniques, weighted mixture minibatches and parameter synchronization, improve the performance of the few-shot object detector in the online learning setting that requires quick responses. Our experiment results show that AirInteraction achieves good object detection performance in diverse object detection datasets, and impressive performance within less than five minutes.

#### ACKNOWLEDGMENT

This work was partially sponsored by the ONR grant #N0014-19-1-2266 and ARL DCIST CRA award W911NF-17-2-0181. The authors would like to thank all members of the Team Explorer for providing data collected from the DARPA Subterranean Challenge.

#### REFERENCES

- [1] A. Borji and L. Itti, "State-of-the-art in visual attention modeling," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 185–207, 2013.
- [2] Q. Zhao and C. Koch, "Learning saliency-based visual attention: A review," *Signal Process.*, vol. 93, no. 6, pp. 1401–1407, 2013.
- [3] P. Isola, J. Xiao, D. Parikh, A. Torralba, and A. Oliva, "What makes a photograph memorable?" *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 7, pp. 1469–1482, 2014.
- [4] M. Markou and S. Singh, "Novelty detection: a review - part 1: statistical approaches," *Signal Process.*, vol. 83, no. 12, pp. 2481–2497, 2003.
- [5] C. Wang, W. Wang, Y. Qiu, Y. Hu, and S. Scherer, "Visual memorability for robotic interestingness via unsupervised online learning," in *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part II*, ser. Lecture Notes in Computer Science, vol. 12347. Springer, 2020, pp. 52–68.
- [6] C. Wang, Y. Qiu, W. Wang, Y. Hu, S. Kim, and S. Scherer, "Unsupervised online learning for robotic interestingness with visual memory," *IEEE Transactions on Robotics*, 2021.
- [7] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common Objects in Context," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2014, pp. 740–755.
- [8] S. Ren, K. He, R. B. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *CoRR*, vol. abs/1506.01497, 2015.
- [9] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*. IEEE Computer Society, 2016, pp. 779–788.
- [10] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. E. Reed, C. Fu, and A. C. Berg, "SSD: single shot multibox detector," in *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part I*, ser. Lecture Notes in Computer Science, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds., vol. 9905. Springer, 2016, pp. 21–37.
- [11] X. Wang, T. E. Huang, J. Gonzalez, T. Darrell, and F. Yu, "Frustratingly simple few-shot object detection," in *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, ser. Proceedings of Machine Learning Research, vol. 119. PMLR, 2020, pp. 9919–9928. [Online]. Available: <http://proceedings.mlr.press/v119/wang20j.html>
- [12] P. J. Silvia, "What is interesting? exploring the appraisal structure of interest," *Emotion*, 2005 Mar;5(1):89-102. doi: 10.1037/1528-3542.5.1.89. PMID: 15755222.
- [13] M. G. Constantin, M. Redi, G. Zen, and B. Ionescu, "Computational understanding of visual interestingness beyond semantics: Literature survey and analysis of covariates," *ACM Comput. Surv.*, vol. 52, no. 2, pp. 25:1–25:37, 2019.
- [14] Z. Bylinskii, P. Isola, C. Bainbridge, A. Torralba, and A. Oliva, "Intrinsic and extrinsic effects on image memorability," *Vision Research*, vol. 116, pp. 165–178, 2015, computational Models of Visual Attention. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0042698915000930>
- [15] M. Gygli, H. Grabner, H. Riemenschneider, F. Nater, and L. V. Gool, "The interestingness of images," *2013 IEEE International Conference on Computer Vision*, pp. 1633–1640, 2013.
- [16] L.-C. Hsieh, W. H. Hsu, and H.-C. Wang, "Investigating and predicting social and visual image interestingness on social media by crowdsourcing," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 4309–4313.
- [17] R. Girshick, "Fast r-cnn," in *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 1440–1448.
- [18] J. Dai, Y. Li, K. He, and J. Sun, "R-fcn: Object detection via region-based fully convolutional networks," in *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, D. D. Lee, M. Sugiyama, U. von Luxburg, I. Guyon, and R. Garnett, Eds., 2016, pp. 379–387.
- [19] K. He, G. Gkioxari, P. Dollár, and R. B. Girshick, "Mask R-CNN," in *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*. IEEE Computer Society, 2017, pp. 2980–2988.
- [20] O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, and D. Wierstra, "Matching networks for one shot learning," in *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, 2016, pp. 3630–3638.
- [21] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, ser. Proceedings of Machine Learning Research, vol. 70. PMLR, 2017, pp. 1126–1135.
- [22] H. Chen, Y. Wang, G. Wang, and Y. Qiao, "LSTD: A low-shot transfer detector for object detection," in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*. AAAI Press, 2018, pp. 2836–2843.
- [23] J. Wu, S. Liu, D. Huang, and Y. Wang, "Multi-scale positive sample refinement for few-shot object detection," in *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XVI*, ser. Lecture Notes in Computer Science, vol. 12361. Springer, 2020, pp. 456–472.
- [24] X. Yan, Z. Chen, A. Xu, X. Wang, X. Liang, and L. Lin, "Meta R-CNN: towards general solver for instance-level low-shot learning," in *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*. IEEE, 2019, pp. 9576–9585.
- [25] B. Kang, Z. Liu, X. Wang, F. Yu, J. Feng, and T. Darrell, "Few-shot object detection via feature reweighting," in *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*. IEEE, 2019, pp. 8419–8428.
- [26] J. Pérez-Rúa, X. Zhu, T. M. Hospedales, and T. Xiang, "Incremental few-shot object detection," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*. Computer Vision Foundation / IEEE, 2020, pp. 13 843–13 852.
- [27] Y. Wang, D. Ramanan, and M. Hebert, "Meta-learning to detect rare objects," in *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*. IEEE, 2019, pp. 9924–9933.
- [28] B. Li, C. Wang, P. Reddy, S. Kim, and S. Scherer, "Airdet: Few-shot detection without fine-tuning for autonomous exploration," *CoRR*, vol. abs/2112.01740, 2021.
- [29] I. Culjak, D. Abram, T. Pribanic, H. Dzapo, and M. Cifrek, "A brief introduction to opencv," in *2012 proceedings of the 35th international convention MIPRO*. IEEE, 2012, pp. 1725–1730.
- [30] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, A. Y. Ng, et al., "Ros: an open-source robot operating system," in *ICRA workshop on open source software*, vol. 3, no. 3.2. Kobe, Japan, 2009, p. 5.
- [31] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*. IEEE Computer Society, 2016, pp. 770–778.