Unsupervised Online Learning for Robotic Interestingness with Visual Memory
Chen Wang, Yuheng Qiu, Wenshan Wang, Yafei Hu, Seungchan Kim, and Sebastian Scherer

Abstract—Autonomous robots frequently need to detect “interesting” scenes to decide on further exploration, or to decide which data to share for cooperation. These scenarios often require fast deployment with little or no training data. Prior work considers “interestingness” based on data from the same distribution. Instead, we propose to develop a method that automatically adapts online to the environment to report interesting scenes quickly. To address this problem, we develop a novel translation-invariant visual memory and design a three-stage architecture for long-term, short-term, and online learning, which enables the system to learn human-like experience, environmental knowledge, and online adaptation, respectively. With this system, we achieve an average of 20% higher accuracy than the state-of-the-art unsupervised methods in a subterranean tunnel environment. We show comparable performance to supervised methods for robot exploration scenarios showing the efficacy of our approach. We expect that the presented method will play an important role in the robotic interestingness recognition exploration tasks.

Index Terms—Unsupervised Learning, Online Learning, Visual Memory, Robotic Interestingness

I. INTRODUCTION

I NTERESTING scene recognition is crucial for autonomous exploration, which is one of the most fundamental capabilities of mobile robots [2]. It has a significant impact on decision-making and robot cooperation but has received limited attention. Take the exploration task in Fig. 1 and 2 as an example, finding a narrow passage in Fig. 1 (c) and a door in Fig. 1 (b) and 2 (g) may affect path planning, while the hole in the wall in Fig. 2 (f) may attract more attention for future exploration. However, prior algorithms often have difficulty when they are deployed to unknown environments, as the robots not only have to find interesting scenes but also need to lose interest in repetitive scenes, i.e., interesting scenes may become uninteresting during robot exploration after repeatedly observing similar scenes or objects. For example, a robot may find many things interesting when entering a mine tunnel in Fig. 1 while it should quickly lose interest in the repetitive scenes such as tunnel walls, railways, and headlights, although they could and should receive attention for their first appearance. Nevertheless, recent approaches of interestingness detection [3], [4], saliency detection [5], anomaly detection [6], [7], novelty detection [8], and meaningfulness detection [9] algorithms cannot achieve this type of online adaptability.

To solve this, we propose to establish an online learning scheme to search for interesting sites for robotic exploration. Existing algorithms are heavily dependent on the back-propagation algorithm [10] for learning, which is computationally expensive. To overcome this difficulty, we introduce a novel translation-invariant 4-D visual memory to identify and recollect visually interesting scenes. Since it has no learnable parameters, our algorithm can run very fast. Human beings have a great capability to guide visual attention and judge the interestingness [11]. For mobile robots, we found the following two properties necessary to establish a sense of visual interestingness.

a) Unsupervised: As shown in Fig. 2 interesting scenes in robotic operating environments are often unique and unknown. Therefore, the labels are normally difficult to obtain, but prior research mainly focused on supervised methods [11], [12] and suffers from prior unseen environments. In this paper, we hypothesize that a sense of interestingness can be established in an unsupervised manner for mobile robots.

b) Task-dependent: In practical applications, we often only know uninteresting scenes before a mission is started. Following the example of the tunnel exploration task in Fig. 2 deployment will be more efficient and easier if the robots can be taught what is uninteresting within several minutes. Therefore, we argue that the visual interestingness recognition system should be able to learn from negative samples quickly.
so an incremental learning method is necessary. Note that we expect the model to be capable of learning from negative samples, but that is not necessary for all tasks.

To achieve the properties above, we introduce a three-stage learning architecture for robotic interestingness recognition:

c) Long-term learning: In this stage, we expect a model to be trained offline with big data in an unsupervised manner. Inspired by that the animals (human beings) acquire new information about the world through mechanisms of learning, and retain the information through mechanisms of memory [14], we expect the model to acquire long-term knowledge from their experiences (big data) and retain the knowledge by freezing learnable weights. We expect the training time on a single machine to be on the order of days.

d) Short-term learning: For task-dependent knowledge, the model should be able to learn from hundreds of uninteresting images in minutes. This can be done before a mission is started and is beneficial to quick robot deployment.

e) Online learning: During mission execution, the system should express the top interests in real-time. Similar to the learning process of human beings, which is a special culture that provides continuous development of consciousness in ontogenesis [15], the interestingness should also be updated online, e.g., the interest in repetitive scenes should be lost, even if they are not seen in the short-term learning.

Another important aspect of online learning is that no data leakage is allowed, i.e., each frame is processed without using information from its subsequent frames. This is in contrast to prior works [3], [4] and datasets [16], where interesting frames can only be selected after the entire sequence is processed [17]. Since robots need to respond in real-time, we expect that our algorithms should be capable of adapting quickly. To better measure the capability of online response, we will also introduce a new evaluation metric.

In summary, our contributions are:

- We introduce a simple yet effective three-stage learning architecture for robotic interesting scene recognition, which is crucial for practical applications. We leverage long-term learning to acquire human-like experience, short-term learning for quick robot deployment and acquiring task-related knowledge, and online learning for environmental adaptation and real-time response.

- To accelerate the short-term and online learning, we offer a novel 4-D visual memory to replace the back-propagation algorithm. We introduce cross-correlation similarity for translational invariance, which is crucial for perceiving video streams. We also initiate a tangent operator for safe memory writing, which is crucial for incremental learning from negative samples for fast deployment.

- To measure the online performance, we introduce a strict evaluation metric, i.e., the area under the curve of online precision (AUC-OP) to jointly consider precision, recall rate, and online performance.

- It is demonstrated that our approach achieves 20% higher performance than the state-of-the-art algorithms and can find meaningful scenes in practical applications.

A preliminary version of this work was selected for oral presentation [1]. In this journal version, we have substantial improvements, including (1) We provide a mathematical formulation and an overview for our framework in Section III to better describe the problem of robotic interestingness recognition and address the difference between anomaly and saliency detection. (2) We introduce a new balance mechanism, a usage vector, in Section IV-A for the visual memory to use the space more efficiently, which allows us to achieve comparable accuracy but a faster speed. We show in Section VI-D that it achieves 69 FPS compared to 14 FPS in the previous version. This makes our algorithm applicable to ultra-low power systems, which is vital for mobile robots. (3) We add ablation study in Section VII-C to provide an intuitive explanation for the novel memory usage vector. (4) We update the overall performance in Section VI-E1 for the SubT dataset and add comparison with anomaly detection methods including MemAE [18] and FFP [19]. (5) We show that our unsupervised online learning algorithm is even comparable with the supervised method [20] in Section VI-E2. (6) We make the new version of source codes publicly available based on the robot operating system (ROS),
provide visualization tools for interestingness map in Fig. 1
for real-time human inspection, and provide the density map
to indicate the location of interesting objects in Section VI-G.

II. RELATED WORK

A learning system that encodes the three-stage learning
architecture for interesting scene recognition has not been
achieved yet. Consequently, the problem formulation and
performance evaluation will be very different from prior
approaches. Some existing work on interestingness recognition
has different objectives [12], e.g., Shen et al. aimed to predict
human interestingness on social media [21], which is not
suitable for comparison. Therefore, we will also review the
related techniques in saliency, anomaly, and novelty detection,
since some techniques are also useful for this work.

The definition of interestingness is subjective, thus human
annotations are often averaged over different participants. To
mimic human judgment, prior works have paid much attention
to investigate the relationship of human visual interestingness
and image features [12]. They are usually motivated by
psychological cues and heavily leverage human annotations for
training. This has spawned a large family of supervised learning
methods. For instance, Dhar et al. designed three hand-crafted
rules, including attributes of composition, content, and sky-
illumination to approximate both aesthetics and interestingness
of images [22]. Grabner et al. integrated four features inspired
cognitive concepts to predict interestingness, including raw
pixel values, color histogram, HOG feature, and Gist of the
scene [17]. Gygli et al. introduced a set of features computa-
tionally capturing three main aspects of visual interestingness,
i.e., unusualness, aesthetics, and general preferences [23]. Jiang
et al. extended image interestingness to video and evaluated
hand-crafted visual features for predicting interestingness on
the YouTube and Flickr datasets [4]. Fu et al. formulated
interestingness as a problem of unified learning to rank,
which is able to jointly identify human annotation outliers
[24], [25]. Although numerous efforts have been devoted, the
performance of hand-crafted and feature-based methods is still
not satisfactory in unknown environments.

Deep neural networks played more and more significant
roles in recent works on interestingness recognition. For
example, Gygli et al. introduced VGG features [26] and used
a support vector regression model to predict the interesting-
ness of animated GIFs [3]. Chaabouni et al. constructed a
customized CNN model to identify salient and non-salient
windows for video interestingness recognition [27]. Inspired
by a human annotation procedure of pairwise comparison,
Wang et al. combined two deep ranking networks [28] to
obtain better performance. This method ranked first in the
2017 interestingness recognition competition [29]. Shen et
al. combined both CNN and LSTM [30] for feature learning
to predict video interestingness [21] for media content.

However, the aforementioned methods are highly dependent
on human annotation in training, which is labor expensive
and unsuitable for real-time response [11]. Some efforts for
unsupervised learning have been made in [31], where interesting
events of videos are detected using the density ratio estimation
algorithm with the HOG features [32]. Nevertheless, the
approach cannot adapt well to changing distributions.

In the long-term stage, we introduce an autoencoder [33]
for unsupervised learning, which has been widely used for
feature extraction in many applications. For example, Hasan et
al. showed that an autoencoder is able to learn regular dynamics
and identify irregularity in long-duration videos [9]. Zhang
et al. introduced dropout into the autoencoder for pixel-wise
saliency detection in images [5]. Zhao et al. presented a spatio-
temporal autoencoder to extract both spatial and temporal
features for anomaly detection [34].

In order to learn online, we introduce a novel visual memory
module into the convolutional neural networks. Visual memory
has been widely investigated in neuroscience [35]. While in
computer vision, memory-aided neural networks have received
limited attention and are used only for several tasks. For
example, Graves et al. presented differentiable neural Turning
machines (NTM) [36] to couple external memory with recurrent
neural networks (RNN). Later a differentiable neural computer
(DNC) was introduced by adding a mechanism into NTM to
ensure that the allocated memory does not overlap and interfere
[37]. Santoro et al. extended NTM and designed a module to
efficiently access to the memory [38]. Gong et al. introduced
a memory module into an auto-encoder to remember normal
events for anomaly detection [18]. Kim et al. introduced the
memory network into GANs to remember previously generated
samples to alleviate the forgetting problem [39]. However, the
memories in the above works are defined as flattened vectors,
thus the spatial structural information cannot be retained. In
this paper, we present a translation-invariant memory module
and introduce online learning for robotic interestingness.

III. FORMULATION & OVERVIEW

To better describe the problem, we define interestingness
and the task of interestingness recognition as follows.

Definition (Interestingness). In psychology, interestingness is
a power of attracting or holding one’s attention [40]. In robotics,
interestingness is the power of scenes/objects influencing a
robotic mission such as exploration and decision-making.

Intuitively, interestingness is subjective and is often attenu-
atated on repetitive scenes, especially for robotic applications,
where unique scenes have higher probabilities to be interesting.
Hence, we next formulate robotic interestingness recognition
as a task where temporal information plays an important role.

Formulation. In the task of interestingness recognition, we will
observe an ordered image sequence, i.e., \( I_{1:t} = [I_1, I_2, \ldots, I_t] \),
where each image \( I_i \) is associated with a label \( z_i \). It is assumed
that every item \( (I_i, z_i) \) satisfies \( P_I(I_{1:i-1}, z_{1:i-1}) \),
where \( P_I \) is a probability distribution describing the
interestingness learning task at time \( t \). The goal of interestingness
recognition is to learn a predictor \( f(I_t) \) to associate the ordered
image sequence \( I_{1:t} \) with label \( z_{1:t} \) such that
\( (I_t, z_t) \sim P_I \).

It can be seen that the interestingness label \( z_t \) is not only
related to its associated image \( I_t \), but also related to all its
preceding images, hence the order of image sequence \( I_{1:t} \) will
As mentioned before, we retain the long-term knowledge by will further explain their behavior in Section V.

One important motivation is that the memory have higher probabilities to be interesting scenes. Intuitively, the scenes that cannot be well recollected by the memory module in the short-term and online learning in Fig. 3.

In this paper, we focus on unsupervised learning for unknown interests, since supervised models normally perform better for known interests. For example, a pre-trained human detector is often preferred in the mission of search and rescue.

To achieve the expected properties for mobile robots, we also need such an online learning algorithm to run very fast on embedded low-power processors. Therefore, we introduce a simple yet effective three-stage learning architecture in Fig. 3 that consists of a reconstruction and a memory module. We next provide an overview for the three-stage framework and will further explain their behavior in Section V.

The long-term learning stage in Fig. 3 (a) is a reconstruction model serving for the human-like long-term experience. This stage generally requires a good generalization ability and is to mimic the long-term learning behavior of the human brain [14], in which learned knowledge is not easy to forget. As mentioned before, we retain the long-term knowledge by freezing learnable weights in the model. Basically, the model can be any unsupervised model that can learn from experience. In the experiments, we use a simple autoencoder for efficiency.

To identify unique interests, we then incorporate the visual memory module in the short-term and online learning in Fig. 3 (b) and Fig. 3 (c). One important motivation is that the memory module is able to perform fast learning, which makes it very suitable for memorizing and recollecting visited scenes in real-time. In this way, we are able to identify the interests by comparing the current scene and the most similar visited scenes. Intuitively, the scenes that cannot be well recollected by the memory have higher probabilities to be interesting scenes.

Therefore, this is challenging for existing memory modules including [36], [37], since a live video stream often contains a large amount of redundant information, thus we have to improve the space utilization. Another challenge is that the robots often move continuously, which will lead the video to contain many translational movements. This requires the memory recollecting to be invariant to translation. To achieve these properties, we will first define a novel visual memory in Section IV then explain how this memory can be incorporated into the three-stage learning architecture to identify robotic interestingness.

IV. VISUAL MEMORY

Due to limited storage, we are not able to store all the historical images I_t, but it might be possible to memorize their approximation, i.e., visual features, which will be denoted as x_t. To retain the structural information of visual inputs, we define visual memory M as a 4-D tensor, i.e., M ∈ ℝ^{n×c×h×w}, where n is the number of memory cubes (capacity) and c, h, and w are the channel, height, and width of each cube, respectively. A memory has two operations, i.e., writing and reading. Writing is to encode visual inputs into the memory, while reading is to recall the memory regarding the visual inputs.

A. Memory Writing

We first present the memory writing in the previous version [1], then show how we can further improve the space usage based on that. The goal of visual memory is to balance the old knowledge and new knowledge from visual inputs. Denote the visual inputs at time t as x(t) ∈ ℝ^{c×h×w}, we can define the writing protocol for the i-th memory cube M_i at time t as

\[ M_i(t) = (1 - w_i(t)) \cdot M_i(t - 1) + w_i(t) \cdot x(t), \]

where w_i is the i-th element of a weight vector w ∈ [0, 1]^n,

\[ w(t) = \sigma(\gamma_w \cdot \tan(\frac{\pi}{2} \cdot D(x(t), M(t - 1))), \]

where \( \sigma(\cdot) \) is the softmax function and D(x, M) is a cosine similarity vector, in which the i-th element \( D_i(x, M) \) is

\[ D_i(x, M) = \frac{\sum (x \odot M_i)}{\|x\|_F \cdot \|M_i\|_F}, \]

where \( \odot, \sum, \) and \( \|\cdot\|_F \) are element-wise product, elements summation, and Frobenius norm, respectively. The writing
protocol in (1) is a moving average, in which the learning speed can be controlled via the writing rate $\gamma_w$ ($\gamma_w > 0$), so that the training samples can be learned at an expected speed. The effect of writing rate $\gamma_w$ is presented in Section VII-E.

It’s worth mentioning that, to promote the sparsity of memory, we introduce a tangent operator in (2) to map the range of cosine similarity $[-1, 1]$ in (3) to $[-\infty, \infty]$. Therefore, memory writing can be focused on fewer, but more relevant cubes via the softmax function. This leads to easier incremental learning compared to [36], [37], which will be explained in Section V-B and Section VII-A. Note that giving a very low learning compared to [36], [37], which will be explained in Section VI-D.

B. Memory Reading

Recall that convolutional features (visual inputs) are invariant to small translations due to the concatenation of pooling layers to convolutional layers [41]. To obtain invariance to large translations, people often rely on data augmentation, which is very computationally heavy. To solve this problem, we introduce an invariance to translation into the memory reading, leveraging that the structural information of visual inputs is retained in memory writing. Denote 2-D circular translation of the $i$-th memory cube along the horizontal and vertical directions with $(x, y)$ elements at time $t$ as $M^i_{(x,y)}(t)$. The memory reading $f(t) \in \mathbb{R}^{n \times h \times w}$ is defined as

$$f(t) = \sum_{i=1}^{n} r_i \cdot M^i_{(x,y)}(t),$$

where $r_i$ is the $i$-th element of reading weight vector $r \in [0, 1]^n$.

$$r = \sigma(\gamma_r \cdot \tan(\frac{\pi}{2} \cdot S(x(t), M(t)))),$$

where $\gamma_r > 0$ is the reading rate. The $i$-th element of $S(x, M)$ is the maximum cosine similarity of $x$ with $M^i_{(a,b)}$, where $a = 0 : h - 1$ and $b = 0 : w - 1$ imply all translations. Intuitively, to find the maximum cosine similarity, we need to repeatedly compute (3) for translated memory cube $h \times w$ times, resulting in high computational complexity. To solve this problem, we leverage the fast Fourier transform (FFT) to compute the cross-correlation [42]. Recall that 2-D cross-correlation is the inner-products between the first signal and circular translations of the second signal [43], and the numerator of cosine similarity in (3) is also the inner products. We can therefore compute the maximum cosine similarity as

$$S_i(x, M) = \max \frac{\mathcal{F}^{-1}(\sum_c ^\mathcal{S} \hat{x}^c \odot \hat{M}_i)}{\|x\|_F \|M_i\|_F},$$

where $\hat{\cdot}$ is the 2-D FFT, $\cdot^\mathcal{S}$ is the complex conjugate, and $\sum_c$ is element-wise summation along channel dimension. The translation $(x, y)$ in (6) for the $i$-th memory cube is corresponding to the location of the maximum response,

$$(x, y) = \arg \max_{(a,b)} \sum_c ^\mathcal{S} \hat{x}^c \odot \hat{M}_i(a, b).$$

In this way, the computational complexity for each memory cube can be reduced from $O(ch^2w^2)$ to $O(chw \log hw)$. Another advantage of translation-invariance in memory reading is that the memory usage becomes more efficient through sharing the cubes with a translational difference, since scene translation is common in the continuous video streams for many robotic applications, e.g., robot exploration and object search. This will be further explained in Section VII-D.

V. LEARNING

We next present the three-stage learning architecture for visual interestingness that incorporates the memory module into an autoencoder to achieve the important properties mentioned in Section I and [41]. An overview of the architecture is shown in Fig. 3 where each of the stages will be explained in the following sections, respectively.

A. Long-term Learning

Inspired by the fact that humans have a massive storage capacity [44], we use a reconstruction model in Fig. 3(a) for long-term learning due to the following reasons.
a) Unsupervised Knowledge: A reconstruction model can be trained in an unsupervised way, hence we can collect a massive number of images from the internet or in real-time during execution to train the model without much effort. This agrees with the objective of long-term learning that is to memorize as many scenes as possible. In this stage, we still leverage the back-propagation algorithm for training, so a large amount of the knowledge will be ‘stored’ in the learnable parameters, which will be frozen afterward. In this sense, the learned knowledge can be treated as an unforgettable human-like experience, which will be important in the following stages.

b) Detailed and Semantic: To precisely reconstruct images, the feature map in the bottleneck layer needs to contain detailed information. On the other hand, since the size of the feature map is normally much smaller than the input images, it has to contain higher-level semantic information. This is crucial for visual interestingness, since both texture and object-level information may attract one’s interests. Recollect the visual feature \( \mathbf{x}_i \) is an approximated version of image \( \mathbf{I}_i \), we are able to approximate \( P_r \) using the visual memory \( \mathbf{M}(t) \), which will play an important role in the following learning stage.

In this paper, we adopt one of the most simplified models, an autoencoder in Fig. 3 (a), to reconstruct images. We establish the encoder following the architecture of VGG-16 [26] and concatenate 5 deconvolutional blocks [45] for the decoder. If we tried more recent architectures including ResNet [46] and MobileNetV2 [47] and found the performance is less sensitive to small translations due to the architecture of CNNs. We leverage the invariance of visual memory to large translations in short-term and online learning.

B. Short-term Learning

As aforementioned, we often only know the uninteresting scenes before a robotic mission is started. For known interesting objects, we prefer to use supervised object detectors. Therefore, we expect that our unsupervised model can be trained incrementally with negative labeled samples within several minutes. This will be helpful for the robots to learn environment-related knowledge and quick robot deployment.

To this end, we develop a short-term learning architecture in Fig. 3 (b). The memory module \( \mathbf{M} \) is inserted into the trained reconstruction model, in which all parameters are frozen. For each sample, the output of the encoder is first written into the memory, then memory reading is taken as inputs of the decoder. Intuitively, the images cannot be reconstructed well initially, as feature maps are not fully learned by the memory, and memory reading will be different from the encoding outputs. Therefore, we can inspect the reconstruction error to find whether the memory has learned to encode the training samples or not. In the experiment, we used the mean square error (MSE) loss for both long-term and short-term learning. In experiments, we find that the final performance is insensitive to the number of epochs in short-term learning. Therefore, to speed up the deployment process, we early stop short-term learning when either one of the two criteria is satisfied: (1) Visual cues: The decoder provides visually good reconstruction; (2) The MSE loss of the reconstruction is not reduced for 3 epochs. More details about the sensitivity to the short-term learning is presented in Section VI-F4 and Fig. 7.

The memory learning is very fast since it has no learnable parameters. It also has several advantages. Recall that the gradient descent algorithms cannot be directly applied to neural networks for incremental learning, since all trainable parameters are changed during training, leading the model to be biased towards the augmented data (new negative labeled data), resulting in the forgetting of previously learned knowledge. This phenomenon is also called “catastrophic forgetting” in continual learning [48]. Although we can fine-tune the model on the entire data, which takes the learned parameters from long-term learning as an initialization, it is too computationally expensive and cannot meet the requirements for short-term learning. Nevertheless, memory learning is able to solve this problem inherently. One reason is that we introduce the tangent operator in [27] to promote writing sparsity, hence fewer memory cubes are affected, resulting in safer and faster incremental learning. The effects of sparse writing and short-term learning will be shown in Section VI-F2 and Section VI-F4 respectively.

C. Online Learning

Online learning is one of the most important capabilities for a real-time visual interestingness recognition system, as human feelings always keep changing according to environments and experiences. Moreover, people tend to lose interest when repeatedly observing the same objects or exploring the same scenes, which is very common in a video stream from a mobile robot. Therefore, we aim to establish such an online learning architecture for real-time robotic systems in Fig. 3 (c), where only the frozen encoder and memory are involved.

Different from short-term learning, in online learning, the inputs are video stream and memory reading is performed before writing. Intuitively, if unknown scenes or objects appear suddenly, the confidence of memory reading will become lower than before, which can be treated as a new interest. Since the new scenes or objects are then written into the memory, their reading confidence level will become higher in the subsequent images. Therefore, the model will learn to lose interest in repetitive scenes once the scene is remembered by the memory. In this sense, the visual interestingness distribution \( P_r(\mathbf{I}_{1:t}, \mathbf{z}_{1:t}) \) should be negatively correlated with the memory reading confidence. In practice, we adopt averaged (over feature channel) cosine similarity of the memory reading and visual inputs to approximate the reading confidence.

A large translation often happens during robot exploration, hence an invariance to large translations introduced in (6) is able to further reduce memory consumption and improve the system robustness. This will be demonstrated in Section VI-F3 and Section VII-D. Instead of selecting interesting frames after processing an entire sequence [11], we need a few control variables that can be easily adjusted for different applications. For example, a hyper-parameter writing rate \( \gamma_w \) controlling the rate of losing interest will be useful for objects search. This will be further demonstrated in Section VII-E.
VI. Experiments

We will first introduce a new metric to better evaluate the performance of online learning, then the performance on two difficult datasets will be presented. The effects of online learning, writing sparsity, translational invariance, and short-term learning will also be presented, respectively.

A. Evaluation Metric

Prior research typically only focused on the precision or recall rate and is not able to capture the online response of interestingness. Moreover, data leakage often happens in prior methods [3], [4] and datasets [16], where interesting frames are selected only after the entire sequence is processed [17]. To solve the problems, we introduce a new metric, i.e., \( \text{area under the curve of online precision (AUC-OP)} \) to evaluate one frame without using the information from its subsequent frames.

This metric is stricter and jointly considers online response, precision, and recall rate. Intuitively, if \( K \) frames of a sequence are labeled as interesting in the ground truth, an algorithm is perfect if its top \( K \) interesting frames are the same as the ground truth. Considering a sequence \( I_{1:N} \), we take an interestingness prediction \( f(I_t) \) as a true positive (interesting) if and only if \( f(I_t) \) ranks in the top \( K_{t,n} \) among a subsequence \( p(I_{t-n+1}: t) \), where \( K_{t,n} \) is the number of interesting frames in the ground truth. Note that the subsequence \( I_{t-n+1}: t \) only contains frames before \( I_t \), as data leakage is not allowed in the online response. Therefore, we may calculate an online precision score for the subsequences (with length \( n \))

\[
s(n) = \frac{\sum TP}{\sum TP + \sum FP}.
\]

where TP and FP denote the number of true positives and false positives, respectively. Note that all true positives rank in the top \( K_{t,n} \), which means that no false negative is allowed. Recollect that a recall rate can be calculated as \( r = \frac{TP}{TP+FN} \), where FN is the number of false negatives. This means that the proposed online precision score \( s(n) \) requires a 100% recall rate. For a better comparison, we often accept true positive predictions with ranking in the top \( \delta \cdot K_{t,n} \), where \( \delta \geq 1 \). Therefore, the overall performance that jointly considers online performance, precision, and recall rate is the area under the curve of the online precision \( s(\frac{N}{n}, \delta) \), where \( \frac{N}{n} \in [0, 1] \). This includes all subsequences since length \( n \in [1, N] \), hence it can be written as an integral

\[
T(\delta) = \int_0^1 s(x, \delta) dx, \quad \text{(AUC-OP)}
\]

where \( x = \frac{n}{N} \). In practice, we often allow some false negatives and \( \delta = 2 \) is recommended for most exploration tasks. Note that most robotic systems require real-time response, thus their performance on \( \text{AUC-OP} \) is vital. However, the evaluation of real-time response is often ignored by existing works.

B. Implementation

Our algorithm is implemented in the robot operating system (ROS) using the PyTorch library [19]. The memory is initialized with a uniform distribution described in [50]. The reading and writing rates are set as \( \gamma_r = \gamma_w = 5 \). In the previous version [1], a memory capacity of 1000 is adopted. In this paper, we can use a smaller memory capacity of 100 to achieve a similar performance due to the improved efficiency of space usage.

C. Dataset

In long-term learning, we perform unsupervised training with the COCO dataset [51]. In short-term and online learning, we will test our online robotic interestingness recognition system on challenging environments, including the underground and air scenarios. They are the SubT dataset [13] for the unmanned ground vehicles (UGV) and the Drone Filming dataset [52] for unmanned aerial vehicles (UAV). The SubT dataset will be used for quantitative evaluation, while the Drone Filming dataset will be mainly for qualitative evaluation.

The SubT dataset is based on the Defense Advanced Research Projects Agency (DARPA) Subterranean Challenge (SubT) Tunnel Circuit. In this challenge, the competitors are expected to build robotic systems to autonomously search and explore the subterranean environments. The environments pose significant challenges, including a lack of lighting, lack of GPS and wireless communication, dripping water, thick smoke, and cluttered or irregularly shaped environments. Each of the tunnels has a cumulative linear distance of 4-8 km. The dataset listed in Table I contains seven videos (1h) recorded by two fully autonomous UGV during the competition from Team Explorer [53], who won the first place [54] at this event. Each sequence is evaluated by at least three volunteers (robotics engineers and researchers), who are expected to select the frames that could be interesting for a robotic mission. The examples include (a) a scene/object appears for the first time; (b) a scene/object has a sudden change; (c) objects and scenes are mismatched; (d) scenes or objects appear in a new viewpoint. It can be seen in Table I that the SubT dataset is very challenging, as human annotation varies a lot, i.e., only 15.5% and 3.6% of the frames are labeled as interesting by at least one and two subjects, respectively. Since predicting the category with fewer samples is more difficult, we name the two categories in Table I as normal and difficult, respectively.

The Drone Filming dataset is recorded by quadcopters during autonomous aerial filming [52] and publicly available [55]. It also contains challenging environments including intensive light changes, severe vibrations, and motion blur, etc. Different from other scenarios such as surveillance cameras in anomaly

| Sequence | Length | Normal | Difficult |
|----------|--------|--------|-----------|
| I        | 53.1 min | 11.11% | 2.76%     |
| II       | 55.7 min | 15.07% | 4.49%     |
| III      | 79.4 min | 9.37%  | 3.02%     |
| IV       | 80.0 min | 17.51% | 4.29%     |
| V        | 59.0 min | 24.52% | 4.07%     |
| VI       | 57.5 min | 22.77% | 3.30%     |
| VII      | 83.0 min | 11.04% | 3.21%     |
| Overall  | 467.7 min | 15.49% | 3.58%     |

“Normal” and “Difficult” indicates that the percentage of frames that are labelled as interesting by at least 1 or 2 subjects, respectively.

VII 83.0

VI 57.5

IV 80.0

III 79.4

II 55.7

I 53.1

min

min

min

min

min

min

min
Fig. 4. Examples of the detected interesting scenes by our unsupervised online learning algorithm.

### TABLE II
**Runtime of Different Learning Stages.**

| Method     | Short-term Learning | Online Learning |
|------------|---------------------|-----------------|
| [1]†       | 10 min              | 72.01 ms/frame  |
| Ours       | 3 min               | **14.58** ms/frame |

†[1] is the conference version of this work.

Detection, robotic visual systems pose extra challenges due to fast background changes, limited computational resources, and dangerous operating environments to which human beings cannot get access. We next show that our method is suitable for these scenarios and even better than supervised methods.

### D. Efficiency

We conducted the efficiency test on a single Nvidia GPU of GeForce GTX 1080Ti. In long-term learning, it takes about 3 days to perform unsupervised training on the COCO dataset [51]. The running time for short-term and online learning is presented in the Table II, together with the previous version of this work [1]. For short-term learning, our model only takes about 3 minutes to learn 912 uninteresting images in the SubT dataset, which is feasible for deployment purposes of most practical applications. For online learning, it runs about 14.58 ms per frame, which is feasible for a real-time interestingness recognition system (Real-time means as fast as a human brain, i.e., 100 ms/frame [56]). It can be seen in Table II that we achieve a much faster running speed than the previous version. The main reason is that we introduce the usage vector [4] to dramatically improve the efficiency of space usage so that we can use the smaller memory capacity of $n = 100$ instead of 1000 to achieve similar performance. This is important for mobile robots since it makes our method applicable to ultra-low power processors. We further provide an intuitive explanation for the effects of usage vector in Section VII-C.

### E. Performance

To the best of our knowledge, robotic visual interestingness recognition is currently underexplored and existing methods have poor performance in this scenario. To demonstrate the effectiveness of our method, we will compare both unsupervised methods and weakly supervised methods.

1) **Comparison with Unsupervised Methods:** We first compare with the unsupervised methods for anomaly detection, i.e., future frame prediction (FFP) [19] and MemAE [18]. To detect an anomaly, FFP introduces temporal constraint
into video prediction, while MemAE introduces a memory module. The overall performance of [AUC-OP] of our method is presented in Table III. Compared to FFP, our method achieves an average of 2.3%, 5.9%, and 1.8% higher accuracy in the normal category and 6.9%, 3.4%, and 3.6% higher accuracy in the difficult category for $\delta = 1, 2, 3$, respectively. Note that to obtain better performance, we adopt a pre-trained model from FFP [19] and fine-tune it on mixed subterranean videos including Sequence I from the SubT dataset. This might be the reason that FFP performs a little better on Sequence I than our method. Some interesting scenes predicted by our method are presented in Fig. 4, in which it can be seen that many interesting and meaningful scenes, including doors and intersections, are identified correctly.

It can be seen that MemAE [18] has poor performance on the SubT dataset, which is just a little higher than our memory model without translational invariance (w/o invariance). This is not surprising since MemAE is designed for anomaly detection, but not interestingness recognition. One main difference is that video stream in anomaly detection is mostly from a static surveillance camera, thus the background doesn’t change, while in robotic interestingness recognition, there are many fast translational movements. Moreover, MemAE cannot perform online learning to adapt to challenging repetitive environments, hence many false positives are produced.

To better illustrate the evaluation, we next report the performance using other widely-used metrics, including precision and the area under the curve (AUC) of receiver operating characteristic (ROC). The averaged performance on all sequences is listed in Table IV. It can be seen that our method still achieves the best performance. Specifically, our method achieves 1.5% and 5.5% higher performance in precision and 3.7% and 1.5% higher performance in AUC-ROC than FFP in the normal and difficult categories, respectively. This indicates that our method is still effective even using the traditional metrics.

2) Comparison with Weakly Supervised Methods: We next present the comparison with a weakly supervised method for place recognition, NetVLAD [20]. It is inspired by the image representation of vector of locally aggregated descriptor (VLAD) [57], that commonly used in image retrieval. The VLAD layer is pluggable into any CNN architecture and has a good generalization ability. It has been widely used in many place recognition methods such as LPD-Net [58] and PCAN [59]. In the experiments, we adopt the architecture of VGG-16 [26] as its feature extractor and train the network with triplet loss [60] using the Adam optimizer [61], where a learning rate of 0.0001 is adopted. We take the first sequence of the SubT dataset for training, while test on all the other sequences. The training process takes about 2 days to converge.

For fair comparison, we only list the overall performance on the test Sequence II to VII in Table V using the metric of [AUC-OP]. It is not surprising that the supervised method NetVLAD outperforms our method in some cases. However, our method is trained online without labels and still achieves an average of 6.4% and 8.0% higher precision in the most strict case of $\delta = 1$, respectively. This further verifies the effectiveness and efficiency of our method, especially considering that NetVLAD requires a long time in short-term learning, needs lots of human efforts for annotation, and has to be pre-trained.
in similar test environments.

F. Effect of Different Modules

We next present the effects of the proposed modules and learning stages including online learning, writing sparsity, translational invariance, and short-term learning.

1) Effect of Online Learning: Online learning is able to remove many repetitive scenes, thus it can reduce the number of false positives. The curve of online precision for the normal category for \( \delta = 1, 2, \) and \( 3, \) respectively, as seen in the second row of Table VI, in which our model achieves an average of 10%-20% higher overall performance than the one without online learning in both categories.

2) Effect of Writing Sparsity: To show its effectiveness, we replaced our writing protocol with the one used in [36], [37], which doesn’t introduce the tangent operator and is denoted as ‘without (w/o) sparsity’ in the first row of Table VI. Our model achieves an average of 21.7%, 20.7%, and 11.1% higher performance than the model without our sparsity operator in the normal category for \( \delta = 1, 2, \) and \( 4, \) respectively. In the difficult category, our model still has a higher overall performance of 15.5%, 21.5%, and 24.0% for \( \delta = 1, 2, \) and \( 4, \) respectively, which further verifies the effectiveness of the sparsity operator.

3) Effect of Translational Invariance: Without the invariance to large translations, the performance could drop a lot, as translational movement is very common in robotic applications. As can be seen in the fourth row of Table VI, in which our algorithm achieves an average of 32.4%, 33.0%, and 20.5% higher performance than the one without translational invariance (w/o invariance) in the normal category for \( \delta = 1, 2, \) and \( 4, \) respectively. In the difficult category, it achieves a higher margin of 20.3%, 32.4%, and 38.4% for \( \delta = 1, 2, \) and \( 4, \) respectively. We notice that the performance gain has the largest margin compared to other cases, which further verifies the necessity of the introduced translational invariance.

4) Effect of Short-term Learning: Short-term learning plays an important role in quick robot deployment. The performance can be largely improved if some uninteresting scenes are known before a mission. It can be seen in the fourth row of Table VI that our model achieves an average of 14.6%, 12.9%, and 4.4% higher accuracy than the one without short-term learning (w/o short-term) in the normal category for \( \delta = 1, 2, \) and \( 4, \) respectively. Similarly, it achieves a higher margin of 8.6%, 13.8%, and 14.2% in the difficult category, respectively. The short-term learning improves the performance only at the cost of hundreds of negative-labeled samples and several minutes of training time, which verifies the flexibility of this pipeline.
SubT Drone Filming

Fig. 6. Selected density maps for the SubT and Drone Filming datasets. Interesting objects are correctly identified when their first appear.

![The effects of short-term learning](image)

Fig. 7. The final online performance is insensitive to the number of epochs in short-term learning, which indicates its efficiency and flexibility.

As mentioned in Section V-B, we early stop the short-term learning based on either of the visual quality and the loss of reconstruction to speed up the robot deployment, hence the sensitivity of final performance (online learning) to the number of training epochs in short-term learning is also an important indicator of performance. To illustrate this effect, we show the performance of AUC-OP for models with different short-term learning time in Fig. 7. It can be seen that the model performance is stable with respect to the epoch number, which indicates that the memory learning is fast, and the short-term learning generalizes well to the final performance. In the experiments, we find that the model generally produces the best trade-off performance for 3 epochs, while the performance won’t be reduced dramatically if we early stop it after 1 epoch, which is able to speed up the deployment process.

G. Density Map

Recall that in online learning, the memory reading confidence is low when new objects or scenes appear suddenly because they cannot be recollected by the memory. In this sense, we are able to draw a difference map (density map) for the recalled scenes, so the interesting objects may be able to be masked by pixels with high differences. Some examples from the SubT and Drone Filming datasets are presented in Fig. 6, where the map colors are normalized for better visualization. It can be seen that some interesting objects (seen by the first time) are successfully localized such as the surveillance camera, survivor, ambulance, cyclist, etc. Note that in this paper, we don’t focus on the localization of those objects, but only focus on the interestingness of an entire frame. We cannot guarantee that the interesting objects will always be masked by the differences of memory reading, as we expect that the interest in repetitive objects to be lost online. In practice, we take the density maps as a secondary indicator during mission execution.

| Metric   | AUC-PR† | Precision |
|----------|---------|-----------|
| MemAE    | 0.9070  | 0.7694    | 0.9066    |

†The Area Under the Curve of Precision-Recall.

H. Generalization Ability of Visual Memory

In this section, we test the generalization ability of the proposed visual memory by applying it to an underwater robot exploration task. The robots in this task are expected to
collect scientifically relevant data such as coral reef images for environmental monitoring. To this end, we conduct the short-term learning on sand images from the Scott Reef 25 dataset [62], which is recorded by the Sirius autonomous underwater vehicle (AUV). The trajectory densely covers an area of 75m × 50m and consists of 50 parallel tracks and one perpendicular path across the center of the path.

Since this dataset only requires classification and doesn’t require online adaptation, we remove the memory writing during online learning and report the averaged precision at a recall rate of 50% and the area under the curve of precision-recall (AUC-PR) in Table VII together with another memory-based method, MemAE [18]. Our method produces a higher performance of 14.75% and 13.72% than MemAE in terms of AUC-PR and precision, respectively. We also show several examples of the sand and coral reefs detected by our model in Fig. 3. It can be seen that our memory-based method can correctly select coral reef images against the sand images, which verifies its generalization ability.

VII. ABLATION STUDY

In this section, we further test our algorithm and provide intuitive explanations for the influences of the writing protocol, memory capacity, translational invariance, and the ability to lose interest. Following the ablation principle, all configurations in this section are kept the same unless stated otherwise.

A. Writing Protocol

It has been pointed out that memory learning is highly dependent on the writing vector in (2), where a tangent operator is introduced for writing sparsity. This section explores its effect and compares it with the writing vector (11) used in [36], [37]. Note that memory defined in [36] is vectors, thus it is not invariant to large translation. Following the ablation principle, we use the same 4-D memory structure and the translation-invariant reading protocol proposed in this paper:

$$\mathbf{w} = \text{softmax}(\gamma \cdot D(\mathbf{x}, \mathbf{M})),$$

(11)

where $\gamma$ is a parameter related to temperature. To show the writing performance, we write two random 3-D tensors into the memory, i.e., $\mathbf{f}_1$ and $\mathbf{f}_2$, and compare their reading accuracy in terms of the cosine similarity defined in (12).

$$S^c(\mathbf{r}, \mathbf{f}) = \frac{\sum (\mathbf{r} \odot \mathbf{f})}{||\mathbf{r}||_F \cdot ||\mathbf{f}||_F},$$

(12)

where $\mathbf{r}$ and $\mathbf{f}$ are the memory reading and writing tensors consecutively. In experiments, we set $\gamma_w = \gamma_r = 5$ and write both $\mathbf{f}_1$ and $\mathbf{f}_2$ 5 times continuously and show their reading accuracy in terms of number of writing in Fig. 9.

It can be seen that both memories are able to remember the random tensors after repeatedly writing. However, when writing a vector without a tangent operator, the accuracy of $\mathbf{f}_2$ keeps dropping even when $\mathbf{f}_1$ is learned, i.e., $S^c(\mathbf{r}_1, \mathbf{f}_1) \approx 1$. This is because all memory cubes are affected due to the non-sparse writing vector in (11). This will be a severe issue when a robot keeps learning the same thing (observing the same scene), since the learned knowledge may be forgotten due to the non-sparse writing. Nevertheless, our writing vector with the tangent operator is able to map the weight of scene, since the learned knowledge may be forgotten due to the non-sparse writing. Nevertheless, our writing vector with the tangent operator is able to map the weight of $\mathbf{f}_1$ to infinite when $\mathbf{f}_1$ is learned, resulting in safer memory writing as only a few memory cubes are affected. This further verifies the effectiveness of the new writing vector.

B. Memory Capacity

This section explores the effects of memory capacity, i.e., the number of memory cubes $n$, which is an important hyper-parameter for incremental learning. Similarly, we write two same random 3-D tensors $\mathbf{f}_1$ and $\mathbf{f}_2$ 5 times sequentially into two different memories in terms of the memory capacity.

As can be seen in Fig. 10 both memories are able to learn random samples in a short time, while the reading accuracy of $\mathbf{f}_1$ drops a lot for smaller capacity after starting to write $\mathbf{f}_2$, although it is remembered later when $\mathbf{f}_1$ is written again. We observe a similar phenomenon when the number of samples is around the same or larger than the memory capacity. This means that memory with a small capacity quickly forgets old knowledge when learning new knowledge. While for larger capacity, reading accuracy can be less affected by new knowledge, resulting in safer and easier incremental learning, which is vital for real-time robotic applications. We can leverage
We next test it on the Drone Filming dataset \[52\] in Fig. 12. In contrast to ultra-low power processors, e.g., UA V, recall the memory for the 1st frame, since the ambulance is not seen before. However, the module WTI is able to recall the memory precisely in the subsequent frames, while the module WOTI quickly fails, although its reading is still meaningful, e.g., the 2nd and 4th frames have correct patterns for sky, trees, and ground. The recalled memory for the 3rd frame from module WTI is roughly a translated replica of the 2nd frame of the video (this also occurs at the 4th and 5th frame), which means that WTI correctly takes the 2nd frame as the most similar scene to the 3rd frame. This phenomenon verifies the translational invariance of our reading protocol.

Note that there is a small translation between the video and the 2nd frame from WTI. This is because the invariance to small translations of CNN features, i.e., the features look the same for visual memory, although they appear with a small translational difference. Therefore, the introduced cross-correlation similarity together with the CNN features contribute complete invariance of translation to memory recall.

### E. Loss of interest

To test the online learning capability of losing interest for the algorithm, we perform a qualitative test based on the Drone Filming dataset. The objects tracked in this dataset, e.g., cars or bikes, are relatively stable with respect to the drone, while the background of the tracked objects keeps changing due to their movements. This makes it suitable for testing the capability of online learning. One of the video clips is shown in Fig. 13, where two different online learning speeds are adopted, i.e., \( w_s = 1.0 \) and \( w_s = 0.2 \). It can be seen that the interest levels of both settings become high when new objects or scenes appear, which means that both settings are able to detect novel objects. However, the interest level with a larger writing rate always drops faster, meaning that it is quicker to lose interest in similar scenes. This verifies our objective that a simple hyper-parameter can be adjusted for different missions. We notice that in Fig. 10 the memories with different capacities have different learning speeds. It can also be adjusted by the writing rates tested in this section.
VIII. LIMITATION & DISCUSSION

We mainly researched the topic of general interestingness recognition for field robots that often operate in unknown and unstructured environments where interesting objects and scenes are critical for robotic tasks such as search and rescue. However, for other cases such as household robots that often operate in known environments, the potential interesting objects could have different and even multiple definitions in tasks such as cleaning and organizing, where our algorithm may not be applicable. For example, we may be interested in a broom during cleaning, while interested in the entire messy room when reorganizing it. In the future, we will consider the scenario that only specific objects are interesting but not predefined in a robotic mission. To achieve this, we will take users’ commands into account for better adapting to the new environments. The model needs to be updated in real-time and search for similar interesting objects/scenes via online learning from a few labeled examples selected by the user during mission execution. This could further benefit the robotic tasks such as manipulation, exploration, and decision-making.

IX. CONCLUSION

In this paper, we developed an unsupervised online learning algorithm for robotic visual interestingness recognition. We first introduced a novel translation-invariant 4-D visual memory, which can be trained without the back-propagation algorithm. We also introduced a usage vector into the memory to improve the efficiency of space usage. To better fit for practical applications, we designed a three-stage learning architecture: long-term, short-term, and online learning. The long-term learning stage is responsible for human-like, real-life knowledge accumulation and trained on unlabeled data via back-propagation. Short-term learning is responsible for learning environmental knowledge and trained via visual memory for quick robot deployment, while online learning is responsible for environment adaption and leverage visual memory to identify interesting scenes. The experiments show that being implemented on a single machine, our approach is able to learn online and find interesting scenes efficiently in real-world robotic tasks. It is also shown that our approach even outperforms supervised methods with a large margin on the SubT dataset. We expect that it will play an important role in robotic applications such as exploration and decision-making.

ACKNOWLEDGMENTS

This work was partially sponsored by the ONR grant #N 0014-19-1-2266 and ARL DCIST CRA award W911NF-17-2-0181. The human subject survey was approved under #20 19_0000522. The authors would like to thank all members of the Team Explorer for their constructive suggestions and for providing data collected from the DARPA Subterranean Challenge Tunnel Circuit. The authors would also like to acknowledge the Australian Centre for Field Robotics’ marine robotics group for providing the data of Scott Reef 25.

REFERENCES

[1] C. Wang, W. Wang, Y. Qiu, Y. Hu, and S. Scherer, “Visual Memorability for Robotic Interestingness via Unsupervised Online Learning,” in European Conference on Computer Vision (ECCV), 2020.
[2] S. Obwald, M. Bennewitz, W. Burgard, and C. Stachniss, “Speeding-up robot exploration by exploiting background information,” IEEE Robotics and Automation Letters, vol. 1, no. 2, pp. 716–723, 2016.
[3] M. Gygli and M. Soleymani, “Analyzing and predicting gif interestingness,” in Proceedings of the 24th ACM international conference on Multimedia. ACM, 2016, pp. 122–126.
[4] Y.-G. Jiang, Y. Wang, R. Feng, X. Xue, Y. Zheng, and H. Yang, “Understanding and predicting interestingness of videos,” in Twenty-Seventh AAAI Conference on Artificial Intelligence, 2013.
[5] P. Zhang, D. Wang, H. Lu, H. Wang, and B. Yin, “Learning uncertain convolutional features for accurate saliency detection,” in Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 212–221.
[6] B. Zhao, L. Fei-Fei, and E. P. Xing, “Online detection of unusual events in videos via dynamic sparse coding,” in CVPR 2011. IEEE, 2011, pp. 3313–3320.
[7] W. Luo, W. Liu, and S. Gao, “A revisit of sparse coding based anomaly detection in stacked rnn framework,” in Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 341–349.
[8] D. Abati, A. Porrello, S. Calderara, and R. Cucchiara, “Latent space autoregression for novelty detection,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 481–490.
[9] M. Hasan, J. Choi, J. Neumann, A. K. Roy-Chowdhury, and L. S. Davis, “Learning temporal regularity in video sequences,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 733–742.
[10] D. E. Rumelhart, G. E. Hinton, R. J. Williams et al., “Learning representations by back-propagating errors,” Cognitive modeling, vol. 5, no. 3, p. 1, 1988.
[11] M. G. Constantin, M. Redi, G. Zen, and B. Ionescu, “Computational understanding of visual interestingness beyond semantics: literature survey and analysis of covariates,” ACM Computing Surveys (CSUR), vol. 52, no. 2, p. 25, 2019.
[12] X. Amengual, A. Bosch, and J. L. De La Rosa, “Review of methods to predict social image interestingness and memorability,” in International Conference on Computer Analysis of Images and Patterns. Springer, 2015, pp. 64–76.
[13] http://theairlab.org/dataset/interestingness.
[14] P. Goel, V. F. Castellucci, S. Schacher, and E. R. Kandel, “The long and the short of long–term memory—a molecular framework,” Nature, vol. 322, no. 6078, pp. 419–422, 1986.
Y. Fu, T. M. Hospedales, T. Xiang, S. Gong, and Y. Yao, “Interestingness
S. Hochreiter and J. Schmidhuber, “Long short-term memory,” in
M. A. Kramer, “Nonlinear principal component analysis using autoasso-
N. Dalal and B. Triggs, “Histograms of oriented gradients for human
Y. Ito, K. M. Kitani, J. A. Bagnell, and M. Hebert, “Detecting interesting
C.-H. Demarty, M. Sjöberg, B. Ionescu, T.-T. Do, M. Gygli, and N. Duong,
S. Wang, S. Chen, J. Zhao, and Q. Jin, “Video interestingness prediction
R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, “Netvlad: Cnn
W. Liu, W. Luo, D. Lian, and S. Gao, “Future frame prediction for anomaly detection–a new baseline,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 6536–6545.
R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, “Netvlad: Cnn architecture for weakly supervised place recognition,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 5297–5307.
Y. Chen, C.-H. Demarty, and N. Q. Duong, “Deep learning for multimodal-based video interestingness prediction,” in 2017 IEEE International Conference on Multimedia and Expo (ICME). IEEE, 2017, pp. 1003–1008.
S. Dhar, V. Ordonez, and T. L. Berg, “High level describable attributes for predicting aesthetics and interestingness,” in CVPR 2011. IEEE, 2011, pp. 1667–1674.
M. Gygli, H. Grabner, H. Riemenschneider, F. Nater, and L. Van Gool, “The interestingness of images,” in Proceedings of the IEEE International Conference on Computer Vision, 2013, pp. 1633–1640.
Y. Fu, T. M. Hospedales, T. Xiang, S. Gong, and Y. Yao, “Interestingness prediction by robust learning to rank,” in European conference on computer vision. Springer, 2014, pp. 488–503.
Y. Fu, T. M. Hospedales, T. Xiang, J. Xiong, S. Gong, Y. Wang, and Y. Yao, “Robust subjective visual property prediction from crowdsourced pairwise labels,” IEEE transactions on pattern analysis and machine intelligence, vol. 38, no. 3, pp. 563–577, 2015.
K. Simonyan and A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition.” in Intellutional Conference on Learning Research, 2015.
S. Chaabouni, J. Benoist-Pineau, A. Zemmari, and C. B. Amar, “Deep saliency: prediction of interestingness in video with cnn,” in Visual Content Indexing and Retrieval with Psycho-Visual Models. Springer, 2017, pp. 43–74.
S. Wang, S. Chen, J. Zhao, and Q. Jin, “Video interestingness prediction based on ranking model,” in Proceedings of the Joint Workshop of the 4th Workshop on Affective Social Multimedia Computing and First Multi-Modal Affective Computing of Large-Scale Multimedia Data. ACM, 2018, pp. 55–61.
C.-H. Demarty, M. Sjöberg, B. Ionescu, T.-T. Do, M. Gygli, and N. Duong, “Mediaeval 2017 predicting media interestingness task,” in MediaEval Workshop, 2017.
S. Hochreiter and J. Schmidhuber, “Long short-term memory,” Neural computation, vol. 9, no. 8, pp. 1735–1780, 1997.
Y. Ito, K. M. Kitani, J. A. Bagnell, and M. Hebert, “Detecting interesting events using unsupervised density ratio estimation,” in Proceedings of 3rd IEEE International Workshop on Analysis and Retrieval of Tracked Events and Motion in Imagery Streams at ECCV2012. Springer, 2012, pp. 151–161.
N. Dural and B. Triggs, “Histograms of oriented gradients for human detection,” in 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR’05), vol. 1. IEEE, 2005, pp. 886–892.
M. A. Kramer, “Nonlinear principal component analysis using autoassociative neural networks,” AIChE journal, vol. 37, no. 2, pp. 233–243, 1991.
Y. Zhao, B. Deng, C. Shen, Y. Liu, H. Lu, and X.-S. Hua, “Spatio-temporal autoencoder for video anomaly detection,” in Proceedings of the 25th ACM international conference on Multimedia, 2017, pp. 1933–1941.
W. Phillips, “On the distinction between sensory storage and short-term visual memory,” Perception & Psychophysics, vol. 16, no. 2, pp. 283–290, 1974.
A. Graves, G. Wayne, and I. Danihelka, “Neural turing machines,” arXiv preprint arXiv:1410.5401, 2014.
A. Graves, G. Wayne, M. Reynolds, T. Harley, I. Danihelka, A. Grabska-Barwińska, S. G. Colmanarejo, E. Grefenstette, T. Ramalho, J. Agapiou et al., “Hybrid computing using a neural network with dynamic external memory,” Nature, vol. 538, no. 7626, pp. 471–476, 2016.
A. Santoro, S. Bartunov, M. Botvinick, D. Wierstra, and T. Lillicrap, “Meta-learning with memory-augmented neural networks,” in Interna
tional conference on machine learning, 2016, pp. 1842–1850.
Y. Kim, M. Kim, and G. Kim, “Memorization precedes generation: Learning unsupervised gans with memory networks,” in the International Conference on Learning Representations (ICLR), 2018.
WordNet 3.0, https://wordnet.princeton.edu/wordnet, Re
defined March 2021.
I. Goodfellow, Y. Bengio, and A. Courville, Deep learning. MIT press, 2016.
C. Wang, “Kernel learning for visual perception,” Ph.D. dissertation, Nanyang Technological University, 2019.
C. Wang, L. Zhang, L. Xie, and J. Yuan, “Kernel cross-correlator,” in Thirty-Second AAAI Conference on Artificial Intelligence (AAAI), 2018, pp. 4179–4186.
T. F. Brady, T. Konkle, G. A. Alvarez, and A. Oliva, “Visual long-term memory has a massive storage capacity for object details,” Proceedings of the National Academy of Sciences, vol. 105, no. 38, pp. 14 325–14 329, 2008.
J. Long, E. Shehhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 3431–3440.
K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.
M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, “Mobilenetv2: Inverted residuals and linear bottlenecks,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 4510–4520.
D. Lopez-Paz and M. Ranzato, “Gradient episodic memory for continual learning,” in Advances in Neural Information Processing Systems, 2017, pp. 6467–6476.
A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, “Automatic differentiation in pytorch,” in Neural Information Processing Systems Workshop, 2017.
K. He, X. Zhang, S. Ren, and J. Sun, “Delving deep into rectifiers: Surpassing human-level performance on imagenet classification,” in Proceedings of the IEEE international conference on computer vision, 2015, pp. 1026–1034.
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 European conference on computer vision. Springer, 2014.
W. Wang, A. Abuja, Y. Zhang, R. Bonatti, and S. Scherer, “Improved generalization of heading direction estimation for aerial imaging using semi-supervised regression,” in 2019 International Conference on Robotics and Automation (ICRA). IEEE, 2019, pp. 5901–5907.
https://www.subtchallenge.com.
https://www.subtchallenge.com/results.html.
M. C. Potter and E. I. Levy, “Recognition memory for a rapid sequence of pictures.” Journal of experimental psychology, vol. 81, no. 1, p. 10, 1969.
H. Jégou, M. Douze, C. Schmid, and P. Pérez, “Aggregating local descriptors into a compact image representation,” in 2010 IEEE computer society conference on computer vision and pattern recognition. IEEE, 2010, pp. 3304–3311.
Z. Liu, S. Zhou, C. Suo, P. Yin, W. Chen, H. Wang, H. Li, and Y.-H. Liu, “Lpd-net: 3d point cloud learning for large-scale place recognition and environment analysis,” in Proceedings of the IEEE International Conference on Computer Vision, 2019, pp. 2831–2840.
W. Zhang and C. Xiao, “Pean: 3d attention map learning using contextual information for point cloud based retrieval,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 12 436–12 445.
V. Balntas, E. Riba, D. Ponsa, and K. Mikolajczyk, “Learning local feature descriptors with triplets and shallow convolutional neural networks.” in BMVC, vol. 1, no. 2, 2016, p. 3.
D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” arXiv preprint arXiv:1412.6980, 2014.
http://marine.acfr.usyd.edu.au/datasets.
C. Wang, J. Yang, L. Xie, and J. Yuan, “Kervolutional neural networks,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 31–40.