Underwater Moving Object Detection using an End-to-End Encoder-Decoder Architecture and GraphSage with Aggregator and Refactoring

Meghna Kapoor
Indian Institute of Technology Jammu
Jammu and Kashmir
meghna@iitjammu.ac.in

Suvam Patra
Manipal Institute of Technology
Manipal, Karnatka
suvampatra8802@gmail.com

Badri Narayan Subudhi
Indian Institute of Technology Jammu
Jammu and Kashmir
subudhi.badri@iitjammu.ac.in

Vinit Jakhetiya
Indian Institute of Technology Jammu
Jammu and Kashmir
vinit.jakhetiya@iitjammu.ac.in

Ankur Bansal
Indian Institute of Technology Jammu
Jammu and Kashmir
ankur.bansal@iitjammu.ac.in

Abstract

Underwater environments are greatly affected by several factors, including low visibility, high turbidity, backscattering, dynamic background, etc., and hence pose challenges in object detection. Several algorithms consider convolutional neural networks to extract deep features and then object detection using the same. However, the dependency on the kernel's size and the network's depth results in fading relationships of latent space features and also are unable to characterize the spatial-contextual bonding of the pixels. Hence, they are unable to procure satisfactory results in complex underwater scenarios. To re-establish this relationship, we propose a unique architecture for underwater object detection where U-Net architecture is considered with the ResNet-50 backbone. Further, the latent space features from the encoder are fed to the decoder through a GraphSage model. GraphSage-based model is explored to reweight the node relationship in non-euclidean space using different aggregator functions and hence characterize the spatio-contextual bonding among the pixels. Further, we explored the dependency on different aggregator functions: mean, max, and LSTM, to evaluate the model's performance. We evaluated the proposed model on two underwater benchmark databases: F4Knowledge and underwater change detection. The performance of the proposed model is evaluated against eleven state-of-the-art techniques in terms of both visual and quantitative evaluation measures.

1. Introduction

Detection of moving objects in a video scene is one of the most fundamental problems in computer vision. Although several surveillance-based techniques are developed for outdoor scenes and very few technologies are developed for underwater applications till the early twenty-first century. Most of the underwater object detection techniques are employed for tracking marine life for estimating the spread of diseases [15] among the marine animals, cracks in oil and gas pipelines [12], drowning detection [14], etc. These applications make it interesting for many underwater surveillance tasks too. Further, state-of-the-art moving object detection algorithms focus on detecting the shape and structure of the object.

The moving object detection task is more complex in the underwater scenario as compared to conventional above water due to the intrinsic properties of water. There are two main factors that affect underwater images greatly. The former includes when the light coming from the objects in the scene is absorbed and scattered due to the suspended particle present in the water, which produces a haze in the underwater scene. The latter is due to the salinity of the water where the optical light gets attenuated due to the difference in viscosity of the water, which creates the color cast problem in the scene. Further, the poor visibility and decolorization in underwater conditions pose challenges for traditional computer vision techniques to accurately analyze underwater images and videos.

Although many of the conventional object detection algorithms are used in underwater surveillance; very few works are reported which are specifically designed to detect underwater moving objects against the underwater challenges including haze, color cast, poor visibility, decolorization, etc. In the state-of-the-art (SOTA) techniques deep convolutional neural networks (CNN) [1, 3, 16] are used to extract the deep features from the underwater image sequences and draw a projection map from RGB color image to a binary classification of the images as object and background. The projections from the encoder to the decoder are non-invertible due to pooling layers. The assumption of symmetricity impediments the extraction of spatial-contextual information among pixels. This motivates to refactor the latent space variables to define the relationship among the nodes that are necessary to preserve the information and minute details of the object. In SOTA techniques, graph convolution network (GCN) [13] is found to be effective in exploring the convolution network in non-euclidean space. The GCN assumes the neighborhood in non-euclidean space and integrates the information using a mean aggregator [2, 30]. We further, broaden the perspective using GraphSage, i.e., the Graph sampling and aggregator. The learning is based on a function from the local neighborhood, and information among nodes is shared using different aggregator functions.

In this article, we propose a simple yet efficient end-to-end hybrid deep learning architecture that uses both deep learning and graph theory for underwater object detection. In the proposed technique, we adhered to the use of a U-Net architecture which is composed of an encoder and a decoder part. The U-Net architecture is designed with a ResNet-50 backbone. Further, the encoder part is connected to the decoder part through a GraphSage technique. In traditional CNNs, the dependency on the kernel's size and the network's depth results in fading relationships of latent space features. Hence, they are unable to procure satisfactory results in complex underwater scenarios. Hence in the proposed scheme, we explored the utilization of refactoring of latent space vectors using GraphSage network. Further, we explored different aggregator functions in GraphSage to check the refactorization of latent space features.

The main contributions of this article are listed below:

- We explored the hypothesis that projections by convolutional neural networks lose information in latent space and utilize refactoring of latent space vectors using a novel refactoring algorithm i.e. GraphSage for moving object detection.
- A novel projection method of high-dimensional latent space variables to graph space using GrapSage is proposed. Here, each element of latent space is projected as a node of an unordered and unstructured graph, and training is done to learn the edge relationships.
- Further, we used different aggregator functions like LSTM, mean, and max to refactor the relationship among the neighboring nodes of latent variables.

The organization of this paper is as follows. Section 2 depicts the discussions on state-of-the-art techniques. The proposed work with the motivation of the same is provided in Section 3. Section 4 describes the experimental results and analysis of the proposed work. The conclusions and future works are provided in Section 5.

2. State-of-the-art Techniques

The main idea of moving object detection is to classify each pixel of an underwater video frame as foreground or background hence perceiving the shape and structure of the object. Based on the study of SOTA techniques we devise underwater object detection techniques into the following sub-categories.

2.1. Statistical methods

Statistical methods were used to statistically model the pixel information and further estimate the parameters with the relative changes in subsequent frames to detect the object's movement. The process of finding the changes in pixel intensity from two consecutive image frames of a video helps in detecting the foreground from the background. Rout *et al.* [25] proposed a method for local change detection to detect underwater moving objects. In the said work, the authors used a difference of 5 frames to detect the local changes. Vasamsetti *et al.* [28] proposed a multi-frame triplet pattern (MFTP) model to detect underwater moving objects. However, the said method failed in the dynamic background condition. Javed *et al.* [10] proposed a robust principal component analysis-based model for moving object detection. The authors decomposed the input data matrix into a low-rank matrix representing the background image and a sparse component identifying the moving objects. Rout *et al.* [26] proposed a spatio-temporal Gaussian-integrated Wronskian model to detect moving objects from a given video scene. The said method considers the background modeling by exploiting the spatial dependency among the pixels in Wronskian framework and multi-temporal background in the temporal direction with a mixture of Gaussian probability density functions. However, considering the underwater challenges the focus has shifted toward deep learning-based methods. The deep features are extracted and given to the decoder to re-project the information to image space passing through a non-linear activation map to infer the moving object from the frame.

2.2. Deep learning based methods

In SOTA, the Encoder-decoder-based deep learning-based methods are popularly used for moving object detection. These methods extract the deep features from underwater video scenes using deep architectures like CNN, transformers, etc in the encoder part of the network. The extracted features contain the object information and are retained during the training of the end-to-end model. Chen *et al.* [3] proposed a model using a novel attention model comprising long short-term memory. The said method is tested on CDnet and may fail to incorporate underwater dynamics. Lin *et al.* [18] proposed a mask RCNN-based method to detect objects in the underwater environment. However, the said method doesn't preserve the minute details of the moving object. Further, Li *et al.* [17] proposed a method for underwater marine life detection using Faster R-CNN. Recently, Bajpai *et al.* [1] proposed a UNet-based model for underwater moving object detection using the ResNet backbone. The proposed methods fail to retain spatial information. Hence, a re-weighting module is expected to restore the connections in latent space. Fan *et al.* [4] proposed a method for multi-scale contextual features using augmentation of the receptive field. The proposed model has a composite connection backbone to deal with the distortion in texture and blurring due to the scattering effect.

2.3. Graph based methods

Recently, graph convolutional neural networks (GCNNs) are found to be effective in various computer vision tasks such as image classification and semantic segmentation. Xu *et al.* [30] proposed a method based on graph learning to extract relevant contextual information from sparse graph structures. To increase spatial awareness, learnable spatial Gaussian kernels performed the graph inference on graphs. Chen *et al.* [2] proposed a combination of semantic segmentation networks for feature extraction on labels and images, and the inferred features were used to initialize the adjacency matrix of the graphs. GCNNs [29] are a natural choice for analyzing irregularly structured input data represented in non-euclidean space. Giraldo *et al.* [6] proposed a graph CNN-based model for moving object detection in complex environments from unseen videos. The said method uses mask R-CNN, motion, texture, and color features to initialize the graph. One of the major disadvantages of the said model is its dependency on handcrafted feature selection. Moreover, the existing state-of-the-art methods are computationally intensive.

3. Proposed Method

We propose an encoder-decoder architecture for underwater moving object detection as shown in Fig: 1. We use a U-net architecture where the left part of the architecture is the encoder part, and the right side of the architecture is the decoder part. As discussed in the previous section, several algorithms are reported in the state-of-the-art techniques for underwater moving object detection. It may be noted that state-of-the-art techniques use CNN architecture to extract the deep features from underwater images. The convolutional layers project the data from the image domain to a higher dimensional latent space. CNNs are not fully connected networks, and the node connections depend on the spatial neighborhoods. The non-euclidean space doesn't preserve the spatial information, which leads to ill-formulated connections in higher dimensional space. As we go deeper, the space becomes non-euclidean, and the information in non-euclidean latent space is loosely connected in terms of spatial relationship.

Fig: 2 column (b-c) depicts an example of two standard SOTA techniques used for underwater moving object detection: ML-BGS [31] and SubSENSEBGS [27]. It can be seen clearly that, both models fail to detect the objects in case of complex backgrounds. The structural information of the object is lost. In a higher dimensional space, the spatial relationships are not maintained as the projection with a convolutional neural network transforms the euclidean space into a non-euclidean space. Hence, the loss of minute details is observed.

To maintain the relationship among the nodes, a refactoring module is required. In the proposed scheme, we have used a combination of deep CNN and GraphSage algorithms. The deep CNN network extracts the spatial information and projects the extracted features in higher dimensional space. A projection from latent space to graph space is made using GraphSage. GraphSage is used as a reweighting module to re-establish the connection between nodes or feature vector elements. A deep decoder projects the information from feature space to image space to detect the moving objects in the scene with spatio-contextual neighborhood information.

We are aware that the underwater complexities are enormous, which include poor illumination, underwater dynamic environments, objects with different shapes and sizes, and cluttered background. GraphSage [7] is a graph CNN method that can handle irregular and unstructured data by updating the node features in a graph and can better deal the underwater uncertainties. Hence, it is expected that the GraphSage method will be better suited for object detection in underwater conditions. Further, to describe the latent space variables we have considered various aggregators for reweighing. We explored different aggregator functions: mean, max, and LSTM over node relations.

3.1. Encoder for Feature Extraction

In the proposed scheme we adhered to the use of a U-Net architecture with a ResNet-50 encoder for the feature ex-

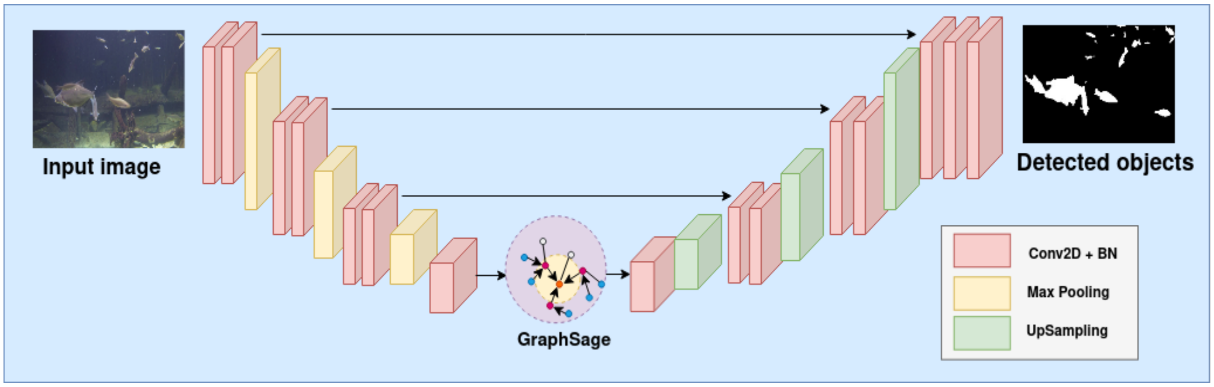


Figure 1. Proposed model for moving object detection using GraphSage

traction. In the ResNet-50 network, CNNs at different levels are used to extract the features from images and project them in higher dimensional space. The deep CNN network extracts the spatial relationship of pixels assuming the information to be in euclidean space. Though convolution projects the information from low-dimensional image space to high-dimensional latent space but unable to preserve the spatio-contextual entity of the image in higher dimensional space. Hence, a feature pooling module or feature reweighting module is required to re-establish the latent space connections. Although several algorithms use feature pooling modules, but are failing to preserve the spatial entity and hence the error in object detection results. Here we propose the use of GraphSage for the same.

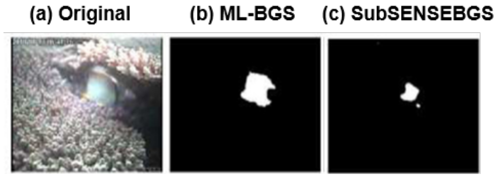


Figure 2. Underwater moving object detection

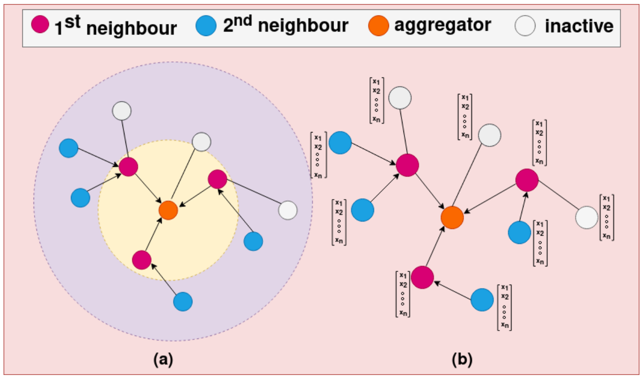


Figure 3. GraphSage Algorithm

3.2. GraphSage

The existing state-of-the-art methods focuses on using graph learning on features extracted by assigning each element as a node of the graph making the archaic methods computationally intensive. However, sampling of graph according to the neighbourhood and then aggregating to devise a relationship is not explored. In our proposed method, the learning strategy of GraphSage is adapted to re-factor the latent feature vector. The latent vector from the feature space is fed to initialize the graph. Every element of the feature vector is considered a node of the graph. Graph architecture tries to find relationships among them. Images have a spatial relationship that can be modeled better in CNN. Using graphs on images leads to high computation time and space. Hence deploying graphs on high dimensional space to refactor the relationship rather than on full image is a better way to get the best of both worlds. GraphSage is used for classification in literature. Liu *et al.* [19] proposed a GraphSage model for forecasting traffic speed. Graphsage is initialized with the historical traffic speeds and geometrical information. Lo *et al.* [20] proposed a GraphSage-based method for intrusion detection. The graph is initialized and trained for edge classification. To the best of our knowledge, no work has been done on node refactorization using GraphSage. To the best of our knowledge, no work has been done on node refactorization using GraphSage.

A graph $\mathcal{G}$ can be defined as an unordered set of tuples defined over vertices ($\mathcal{V}$) and edges ($\mathcal{E}$). The nodes or vertices are connected with links or edges. In the proposed scheme, GraphSage (Graph sample and aggregate) is used for large graphs for inductive reasoning. The basic architecture of the GraphSage algorithm is given in Fig: 3. Further, different stages of the GraphSage are given as follows.

3.2.1 Sampling

The neighborhood $\mathcal{N}$ is defined as the direct hops connected by a pathway as shown in Fig: 3 (a). The neighborhood is defined as a fixed-size subset from the sample set using a uniform draw. The neighbors are updated in each iteration. Working with neighbors helps in reducing the computation time and size. The information from the neighbors is aggregated and given to the node of the next stage.

3.2.2 Aggregator functions

Aggregator functions a_f define the relationship among nodes. The information among the neighboring nodes are shared and updated according to the aggregator function as shown in Fig: 3 (b). In GraphSage, the neighbors (j) in latent space layer l represented by h_j^{l-1} have no order, and the aggregator function represents the particular node while being trainable.

$$h^l_{\mathcal{N}_i} \leftarrow a\_f\{h_j^{l-1} \quad \forall j \in \mathcal{N}(i)\}. \tag{1}$$

In the proposed model, three different aggregator functions are used to re-weight the latent space relationship.
Mean aggregator: The mean of neighborhood nodes is taken into account to evaluate the information at the current node.
Max aggregator: The max or the pooling function operates by doing element-wise max across the neighboring nodes.
LSTM aggregator: Compared to the above two functions, the third is the most complex function and is inherently not symmetric. Random permutation among the neighbors is applied.

3.2.3 Refactoring using GraphSage

Every element of the feature space is considered as a node. The information from neighboring nodes $\mathcal{N}$ is defined over the information from previous nodes. An aggregator function from the set {mean, max, LSTM} is applied over the obtained information from neighbors and is denoted as $h^l_{\mathcal{N}}$. The current information and information from the neighborhood are concatenated. The obtained vector is passed through a non-linear activation (sigmoid in our case). The updated representation of node i in layer l is given as:

$$h^l_i = f_{update}(h^l_{\mathcal{N}_i}, h_j^{l-1}). \tag{2}$$

Here, f_{update} can be simply an aggregator operator or any complex function. We have used the update function as a concatenate operator. The algorithmic representation of the proposed GraphSage scheme is provided in Algorithm 1. A graph $\mathcal{G}$ is initialized using a latent vector and iterated over the k-hop neighborhood. At every iteration, the aggregated information among the nodes is updated to learn the spatial-contextual information among non-euclidean space.

3.3. Decoder

The re-weighted features are mapped using an inverse-mapping function by the decoder. In order to preserve most information, an identical mapping is obtained using U-Net architecture. There are skip connections between the encoder and decoder to preserve the information. The model is initialized using ImageNet data, and later, the weights are updated using the F4Knowledge dataset using transfer learning.

The algorithmic enumeration of the proposed scheme is provided in Algorithm 1

Algorithm 1: Proposed Algorithm for Object Detection

```
Input: RGB video frame
Output: A binary segmented frame
for k = 1 to number of epochs do
    capture frame f
    $b_i \leftarrow f$
    for i = 1 to 3 do
        $c_i \leftarrow conv2d(b_i, kernel)$
        $b_i \leftarrow pool(c_i)$
    $m_i \leftarrow b_i$
    $x_i \leftarrow flatten(m_i)$
    $x_i \leftarrow sigmoid(x_i)$
    Graph Initialization; A graph $\mathcal{G}$, Latent space vector $x_i, i \in \mathcal{V}$ , layers $\mathcal{L}$, neighbourhood function $\mathcal{N}$: $i \rightarrow 2^i$, weight matrices $W^l$ $\forall l = 1 \cdots L$
    $h_i \leftarrow x_i \quad \forall i \in \mathcal{V}$
    for l = 1 to L do
        for $i \in \mathcal{V}$ do
            $h^l_{\mathcal{N}_i} \leftarrow aggregator\_function_l\{h_j^{l-1} \quad \forall j \in \mathcal{N}(i)\}$
            $h^l_i \leftarrow \sigma(W^l.concatenate(h_i^{l-1}, h^l_{\mathcal{N}(i)}))$
    $y_i \leftarrow h^l_i \quad \forall i \in \mathcal{V}$
    $k_i \leftarrow conv2d(y_i)$
    for i = 1 to 3 do
        $u_i \leftarrow upsample(k_i) + b_i$
        $d_i \leftarrow conv2d(u_i)$
    $d_i \leftarrow sigmoid(d_i)$
    $\mathcal{L} = -\frac{1}{N}\sum_{k=0}^{N} t_k * log(\hat{t}) + (1-t_i) * log(1-\hat{t_i})$
    compute gradient
    update weights and bias
```

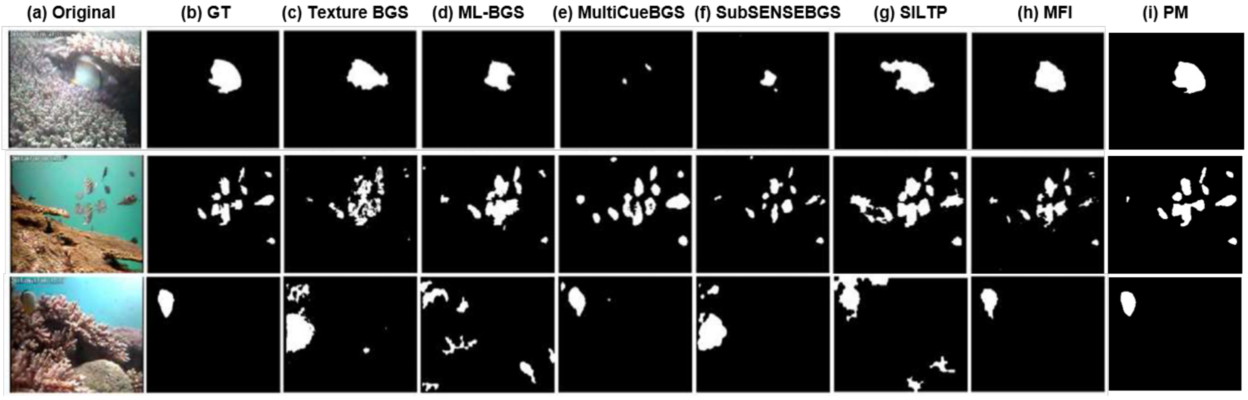


Figure 4. Qualitative measure of F4Knowledge dataset on proposed model

Table 1. Quantitative analysis of proposed method with different aggregator functions on different challenges of F4Knowledge dataset

Challenge	Aggregator Function	Accuracy		Recall	Precision	F measure
		Training	Testing			
ComplexBkg	Mean	99.60	98.59	98.99	99.58	99.28
	Max	99.58	58.77	99.68	59.00	74.13
	LSTM	99.48	99.61	99.03	97.07	98.04
Crowded	Mean	98.83	97.41	99.37	98.02	98.69
	Max	98.88	96.78	99.36	97.39	98.37
	LSTM	98.46	96.96	99.56	97.38	98.46
DynamicBkg	Mean	98.76	97.27	99.27	99.99	99.63
	Max	98.77	97.08	97.08	100.00	98.52
	LSTM	98.75	96.90	97.14	99.75	98.43
Hybrid	Mean	99.14	97.84	98.06	99.78	98.91
	Max	99.13	97.94	98.10	99.84	98.96
	LSTM	99.14	98.66	98.92	99.73	99.32
Standard	Mean	98.74	98.92	99.37	99.54	99.46
	Max	98.73	98.92	99.37	99.54	99.46
	LSTM	98.72	98.29	98.89	99.38	99.13
Aggregate		98.98	95.33	98.81	96.40	97.25

Table 2. Quantitative analysis in terms of F-measure with six SOTA techniques. The red color indicates the best, and blue indicates the second best

Challenge	Texture-BGS [9]	MLBGS [31]	MultiCueBGS [21]	SubSENSEBGS [27]	SILTP [8]	MFI [28]	PM (mean)	PM (max)	PM (LSTM)
complex background	0.69	0.58	0.48	0.21	0.73	0.83	99.28	74.13	98.04
crowded	0.54	0.74	0.68	0.67	0.67	0.69	98.69	98.37	98.46
dynamic background	0.43	0.32	0.33	0.81	0.32	0.64	99.63	98.52	98.43
camouflage foreground	0.42	0.66	0.77	0.42	0.66	0.72	99.46	99.46	99.13
hybrid	0.49	0.46	0.72	0.42	0.69	0.8	98.91	98.96	99.32

Table 3. Quantitative analysis in terms of F-measure with five SOTA architectures. The red color indicates the best, and the blue indicates the second best.

Challenge	GSMM [24]	AGMM [32]	ABMM [11]	ADE [33]	GWFT [22]	PM (mean)
fish swarm	0.57	0.30	0.06	0.59	0.85	0.99
marine snow	0.84	0.82	0.65	0.82	0.91	0.99
small aquaculture	0.77	0.74	0.43	0.88	0.93	0.99
caustics	0.55	0.74	0.67	0.75	0.67	0.99
two fishes	0.79	0.79	0.76	0.71	0.82	0.95

4. Experimental Results and Analysis

The proposed technique is executed on an NVIDIA A100 80 GB GPU with 128 GB RAM. It is implemented by python programming with the PyTorch framework on the Linux operating system. We have evaluated the performance of the proposed scheme on two benchmark databases: F4Knowledge and Underwater change detection. In the proposed scheme, a batch size of 2 is considered during training. In GraphSage, a hop of two neighbors is considered. We used Adam optimizer with a learning rate of e^{-3} to converge our model. The U-Net architecture uses binary cross entropy as a loss function to compute the gradient and update the hyperparameters. The model's performance is tested using different aggregate functions using visual and quantitative evaluation measures. The performance of the proposed scheme is corroborated by comparing its results with those of the eleven state-of-the-art (SOTA) techniques: Texture-BGS [9], MLBGS [31], MLCB [21], Subsense-BGS [27], SILTP [8], MFI [28], GSMM [24], AGMM [32], ABMM [11], ADE [33], GWFT [22].

4.1. Description on Databases

We have evaluated the performance of the proposed scheme on two benchmark databases: F4Knowledge [5] and Underwater change detection [23]. The Fish4Knowledge dataset has video sequences captured from 10 cameras. We considered five challenges from the Fish4Knowledge dataset: complex background, crowded scenes, dynamic background, camouflaged foreground, and hybrid scenes. The number of samples varies between different challenges. The second dataset considered in our experiment is underwater change detection. The said dataset has five videos with different challenges: caustics, marine, fish swarm, two fish, and aquaculture. Fish4Knowledge has less correlation among the frames corresponding to ground truth, while a high correlation among the frames can be observed in the underwater change detection dataset.

4.2. Visual Analysis of Results

The visual analysis of the proposed architecture for underwater moving object detection is carried out on different challenging sequences of the F4Knowledge database and underwaterchangedetection. A visual illustration, of the results on F4Knowledge are shown in Fig: 4 columns (a) and (b) represent the original and ground-truth images of sequences. Fig: 4 columns (c) to (h) represent the results obtained on the considered sequence of the F4Knowldge database using Texture-BGS [9], MLBGS [31], MLCB [21], SubsenseBGS [27], SILTP [8], MFI [28] techniques. It may be observed that most of SOTA methods failed to provide the complete object region. Even many instances the moving object region are missed. However, the results obtained by the proposed scheme as shown in Fig: 4 column (i) are able to detect the object correctly.

Table 4. Quantitative measure on underwater change detection dataset

Challenges	Accuracy	Precision	Recall	F-measure
Caustics	99.61	99.95	99.65	99.80
Marine	98.78	99.80	98.97	99.39
Fish swarm	98.18	99.60	98.51	99.05
Two Fish	98.60	99.38	99.22	99.30
AquaCulture	90.97	93.48	97.36	95.38

4.3. Quantitative Analysis Results

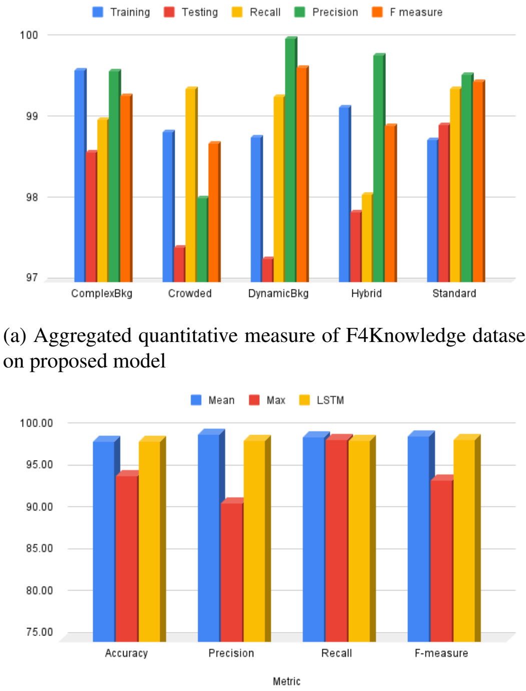
In this article, the evaluation metrics considered to evaluate the quantitative performance of the proposed moving object detection model are accuracy, precision, recall, and f-measure. Accuracy is the ratio of a correctly labeled pixel as foreground among all the pixels. Precision is the ratio of pixels correctly labeled as a foreground to the detected total foreground pixels. The recall is the ratio of pixels labeled as the foreground to those that belong to the foreground. F-measure is the harmonic mean of precision and recall. As the number of background and foreground pixels is not the same, the f-measure is the most reliable metric.

$$F_1 = \frac{TP}{TP + \frac{1}{2}(FP + FN)}. \tag{3}$$

The results obtained by the proposed scheme on the F4Knowledge dataset, using the different aggregator functions are provided in Table: 1 in terms of accuracy, recall, precision, and F-measure quantitative evaluation measures. In this table, we have provided the considered evaluation measures obtained by the proposed scheme on five different challenges of the Fish4Knowledge dataset. The results are found to be very effective and produce a higher accuracy with a very good precision record on all the challenges with different aggregator functions like mean, max, and LSTM. We also observed that for the "complex backgrounds" sequence, the max operator was found to be providing a lesser accuracy.

Further, the proposed model is compared with those of the different state-of-the-art techniques along with considered three aggregator functions: mean, max, and LSTM. The proposed model is compared with the six state-of-the-art techniques: Texture-BGS [9], MLBGS [31], MLCB [21], SubsenseBGS [27], SILTP [8], MFI [28] techniques in terms of F-measure and are shown in Table: 2. It can be clearly observed that the proposed model provides the best results compared to all SOTA techniques. Hence, it corroborates our hypothesis. It is also observed that the mean aggregator surpasses the F-measure as compared to max and LSTM aggregators and other considered SOTA techniques.

We also verified the effectiveness of the proposed scheme on the underwater change detection dataset with the mean aggregator. Table: 3 has quantitative results compared to five SOTA methods: Gaussian switch mixture model (GSMM) [24], adaptive Gaussian mixture model (AGMM) [32], adaptive background mixture model (ABMM) [11], adaptive density estimation (ADE) [33], Gaussian switch with flux tensor (GWFT) [22] in terms of F-measure. Our proposed model was found to be performing best as compared to all the SOTA architectures. Table: 4 contains the quantitative results for the proposed model using the mean aggregator function in terms of accuracy, precision, and recall on five challenges of underwater change detection.



(a) Aggregated quantitative measure of F4Knowledge dataset on proposed model

(b) Comparison of different aggregator functions

Figure 5. Quantitative measure of F4Knowledge dataset on proposed model

4.4. Ablation study

We made an ablation study of the proposed scheme on different aggregators using: mean, max, and LSTM methods on the F4Knowledge dataset which are reported in Table: 5. The comparison of different aggregator functions is reported in Fig: 5. The mean operator has the highest accuracy, precision, and a comparable F-measure with a difference of 0.02 from best. The computation time for the Mean operator is less than LSTM and hence more preferable.

Table 5. Ablation study of different aggregator functions on the F4Knowledge dataset. Red indicates best, and blue indicates second best.

Function	Accuracy	Precision	Recall	F-measure
Mean	98.51	99.38	91.15	98.66
Max	94.46	99.01	98.72	98.66
LSTM	98.50	99.19	93.89	98.68

5. Conclusions and Future Works

In this article, we propose a novel hybrid deep learning and GraphSage architecture for underwater object detection. The proposed model consists of an end-to-end encoder-decoder-based U-Net architecture with the ResNet-50 backbone. To reduce the effects of misclassification in object detection, a novel GraphSage-based model is sandwiched between the encoder and decoder of the U-Net architecture. Three aggregator functions, namely, mean, max, and LSTM, are verified to retain the missing information. The proposed scheme is tested on two benchmark underwater databases: F4Knowledge and underwater change detection. The effectiveness of the proposed scheme is verified with eleven state-of-the-art techniques. It is verified that in non-euclidean space, only convolution operation is insufficient to retain the information. Refactoring the relationship among nodes is necessary. Further, mean based aggregator is found to be providing the best results. In the future, we would like to improve the performance of the proposed scheme using the first generic object neural network tracker for its possible real-time implementation.

References

[1] Vatsalya Bajpai, Akhilesh Sharma, Badri Narayan Subudhi, T Veerakumar, and Vinit Jakhetiya. Underwater U-Net: Deep learning with U-Net for visual underwater moving object detection. In *Proceedings of OCEANS 2021: San Diego–Porto*, pages 1–4, 2021. 2, 3

[2] Shengjia Chen, Zhixin Li, and Zhenjun Tang. Relation R-CNN: A graph based relation-aware network for object detection. *IEEE Signal Processing Letters*, 27:1680–1684, 2020. 2, 3

[3] Yingying Chen, Jinqiao Wang, Bingke Zhu, Ming Tang, and Hanqing Lu. Pixel wise deep sequence learning for moving object detection. *IEEE Transactions on Circuits and Systems for Video Technology*, 29(9):2567–2579, 2019. 2, 3

[4] Baojie Fan, Wei Chen, Yang Cong, and Jiandong Tian. Dual refinement underwater object detection network. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 275–291, Cham, 2020. Springer International Publishing. 3

[5] Robert B Fisher, Yun-Heh Chen-Burger, Daniela Giordano, Lynda Hardman, Fang-Pang Lin, et al. *Fish4Knowledge: collecting and analyzing massive coral reef fish video data*, volume 104. Springer, 2016. 7

[6] Jhony H Giraldo, Sajid Javed, Naoufel Werghi, and Thierry Bouwmans. Graph CNN for moving object detection in complex environments from unseen videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 225–233, 2021. 3

[7] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017. 3

[8] Hong Han, Jianfei Zhu, Shengcai Liao, Zhen Lei, and Stan Z Li. Moving object detection revisited: Speed and robustness. *IEEE Transactions on Circuits and Systems for Video Technology*, 25(6):910–921, 2014. 6, 7

[9] Marko Heikkila and Matti Pietikainen. A texture-based method for modeling the background and detecting moving objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(4):657–662, 2006. 6, 7

[10] Sajid Javed, Thierry Bouwmans, Maryam Sultana, and Soon Ki Jung. Moving object detection on RGB-D videos using graph regularized spatiotemporal RPCA. In *New Trends in Image Analysis and Processing–ICIAP*, pages 230–241. Springer, 2017. 2

[11] Pakorn KaewTraKulPong and Richard Bowden. An improved adaptive background mixture model for real-time tracking with shadow detection. *Video-based surveillance systems: Computer vision and distributed processing*, pages 135–144, 2002. 6, 7, 8

[12] Juhyun Kim, Minju Chae, Jinju Han, Simon Park, and Youngsoo Lee. The development of leak detection model in subsea gas pipeline using machine learning. *Journal of Natural Gas Science and Engineering*, 94:104134, 2021. 1

[13] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016. 2

[14] Fei Lei, Hengyu Zhu, Feifei Tang, and Xinyuan Wang. Drowning behavior detection in swimming pool based on deep learning. *Signal, Image and Video Processing*, pages 1–8, 2022. 1

[15] Daoliang Li and Ling Du. Recent advances of deep learning algorithms for aquacultural machine vision systems with emphasis on fish. *Artificial Intelligence Review*, pages 1–40, 2022. 1

[16] Guohao Li, Matthias Muller, Ali Thabet, and Bernard Ghanem. Deepgcns: Can GCNs go as deep as CNNs? In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9267–9276, 2019. 2

[17] Xiu Li, Min Shang, Hongwei Qin, and Liansheng Chen. Fast accurate fish detection and recognition of underwater images with Fast R-CNN. In *OCEANS 2015 Marine Technology Society/IEEE Washington*, pages 1–5. 3

[18] Wei-Hong Lin, Jia-Xing Zhong, Shan Liu, Thomas Li, and Ge Li. Roimix: Proposal-fusion among multiple images for underwater object detection. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2588–2592, 2020. 3

[19] Jielun Liu, Ghim Ping Ong, and Xiqun Chen. Graphsage-based traffic speed forecasting for segment network with sparse data. *IEEE Transactions on Intelligent Transportation Systems*, 23(3):1755–1766, 2022. 4

[20] Wai Weng Lo, Siamak Layeghy, Mohanad Sarhan, Marcus Gallagher, and Marius Portmann. E-graphsage: A graph neural network based intrusion detection system for iot. In *NOMS 2022-2022 IEEE/IFIP Network Operations and Management Symposium*, pages 1–9, 2022. 4

[21] SeungJong Noh and Moongu Jeon. A new framework for background subtraction using multiple cues. In *Asian Conference on Computer Vision*, pages 493–506. Springer, 2013. 6, 7

[22] Martin Radolko, Fahimeh Farhadifard, and Uwe von Lukas. Change detection in crowded underwater scenes-via an extended gaussian switch model combined with a flux tensor pre-segmentation. In *International Conference on Computer Vision Theory and Applications*, volume 5, pages 405–415. SCITEPRESS, 2017. 6, 7, 8

[23] Martin Radolko, Fahimeh Farhadifard, and Uwe Freiherr von Lukas. Dataset on underwater change detection. In *OCEANS 2016 MTS/IEEE Monterey*, pages 1–8. IEEE, 2016. 7

[24] Martin Radolko and Enrico Gutzeit. Video segmentation via a gaussian switch background model and higher order markov random fields. In *VISAPP (1)*, pages 537–544, 2015. 6, 7, 8

[25] Deepak Kumar Rout, Pranab Gajanan Bhat, T Veerakumar, Badri Narayan Subudhi, and Santanu Chaudhury. A novel five-frame difference scheme for local change detection in underwater video. In *Proceeding of Fourth International Conference on Image Information Processing (ICIIP)*, pages 1–6. IEEE, 2017. 2

[26] Deepak Kumar Rout, Badri Narayan Subudhi, T. Veerakumar, and Santanu Chaudhury. Spatio-contextual Gaussian mixture model for local change detection in underwater video. *Expert Systems with Applications*, 97:117–136, 2018. 2

[27] Pierre-Luc St-Charles, Guillaume-Alexandre Bilodeau, and Robert Bergevin. Flexible background subtraction with self-balanced local sensitivity. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 408–413, 2014. 3, 6, 7

[28] Srikanth Vasamsetti, Supriya Setia, Neerja Mittal, Harish K Sardana, and Geetanjali Babbar. Automatic underwater moving object detection using multi-feature integration framework in complex backgrounds. *IET Computer Vision*, 12(6):770–778, 2018. 2, 6, 7

[29] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 32(1):4–24, 2020. 3

[30] Hang Xu, Chenhan Jiang, Xiaodan Liang, and Zhenguo Li. Spatial-aware graph relation network for large-scale object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9298–9307, 2019. 2, 3

[31] Jian Yao and Jean-Marc Odobez. Multi-layer background subtraction based on color and texture. In *2007 IEEE conference on computer vision and pattern recognition*, pages 1–8. IEEE, 2007. 3, 6, 7

[32] Zoran Zivkovic. Improved adaptive gaussian mixture model for background subtraction. In *Proceedings of the 17th In-*

ternational Conference on Pattern Recognition, 2004. ICPR 2004., volume 2, pages 28–31. IEEE, 2004. 6, 7, 8

[33] Zoran Zivkovic and Ferdinand Van Der Heijden. Efficient adaptive density estimation per image pixel for the task of background subtraction. *Pattern recognition letters*, 27(7):773–780, 2006. 6, 7, 8