PROCEEDINGS OF SCIENCE

Underwater biotope mapping: automatic processing of underwater video data

O. O. Iakushkin,[1,2] E. D. Pavlova,[1,2] A. K. Lavrova,[1] V. V. Polovkov,[1] A. U. Frikh-Khar,[2,4] E. A. Pen,[2] Y. E. Terekhina,[3] A. A. Bulanova,[3] N. V. Shabalin[2] and O. S. Sedova[1]

[1] *Saint-Petersburg State University*
[2] *BioGeoHab*
[3] *Lomonosov Moscow State University*
[4] *Marine Research Center of Lomonosov Moscow State University*

E-mail: Oleg.Jakushkin@gmail.com, e.d.pavlova@spbu.ru

The task of analysing the inhabitants of the underwater world applies to a wide range of applied problems: construction, fishing, and mining. Currently, this task is applied on an industrial scale by a rigorous review done by human experts in underwater life. In this work, we present a tool that we have created that allows us to significantly reduce the time spent by a person on video analysis. Our technology offsets the painstaking video review task to AI, creating a shortcut that allows experts only to verify the accuracy of the results. To achieve this, we have developed an observation pipeline by dividing the video into frames; assessing their degree of noise and blurriness; performing corrections via resolution increase; analysing the number of animals on each frame; building a report on the content of the video, and displaying the obtained data of the biotope on the map. This dramatically reduces the time spent analysing underwater video data. Also, we considered the task of biotope mass calculation. We correlated the Few-shot learning segmentation model results with point cloud data to achieve that. That provided us with a biotope surface coverage area that allowed us to approximate its volume. Such estimation is helpful for precise area mapping and surveillance.

Thus, this paper presents a system that allows detailed underwater biotope mapping using automatic processing of a single camera underwater video data. To achieve this, we combine into a single pipeline a set of deep neural networks that work in tandem.

**** The 6th International Workshop on Deep Learning in Computational Physics (DLCP2022) ****
**** 6-8 July 2022 ****
**** JINR, Dubna, Russia ****

PoS(DLCP2022)024

1. Introduction

Conducting underwater research is necessary to understand better what is on the seabed and whether it can carry out construction work there. Within each state, there are particular documents approving construction plans, as well as approving plans for fishing. Nevertheless, to assess the detailed damage that a particular type of activity will cause to nature, it is necessary to understand: what species of animals and plants live within a kilometre or 500 m around.

At the moment, to solve this problem, measurements are being made: data is taken point by point using video equipment and transmitted to experts, or a unique grid is lowered, which captures part of the soil and lifts it to the ship, and then geologists carry out an assessment.

In the case of considering schools of fish or other non-deep water objects of marine flora and fauna, it must be taken into account that the results of their activities sink and participate in the biological cycle. That is, the characteristic inhabitants of the bottom use them. Knowing which animals and plants are on the seafloor can help us guess which species swim higher.

The main task and solution in this context are watching many hours of video. For each object, measurements are made at specific points, according to which a reporting table is compiled, consisting of a list of living beings recorded by an expert, indicating time intervals on video—the use of platforms for labelling and aggregating data, such as «Yandex.Toloka» is impossible since professional training is required to recognise underwater biological diversity [1].

Solving the problem of automating the work of experts in the field of biotope analysis and mapping by employing artificial intelligence requires the system to be able to detect and identify various underwater objects and traces of their vital activity in low visibility conditions [2]. In addition to information from photographic images (video streams), the system may have information generated from a video about the depth and topography of the area where the shooting was carried out. Thus, the system has an additional context within which the biotope is recognised.

2. Problem definition

Viewing many hours of video recordings can take an enormous amount of time for experts since it is necessary to fix the number of objects of each type at time intervals of 10 seconds. At the same time, over time, the accuracy of the assessment may decrease since the frames contain elements of the biotope of different sizes, which, together with blurring and distortion of the image [3], harms perception.

From the point of view of applied machine learning, it is necessary to consider the presence of a significant difference in the size of the objects of interest to achieve high prediction accuracy. Another critical factor is the lack of a large amount of labelled data from a particular area (White Sea).

Thus, to automate the analysis of underwater video sequences, it is necessary to solve the problems of classification and segmentation in the absence of labelled data and take into account the specifics of underwater shooting.

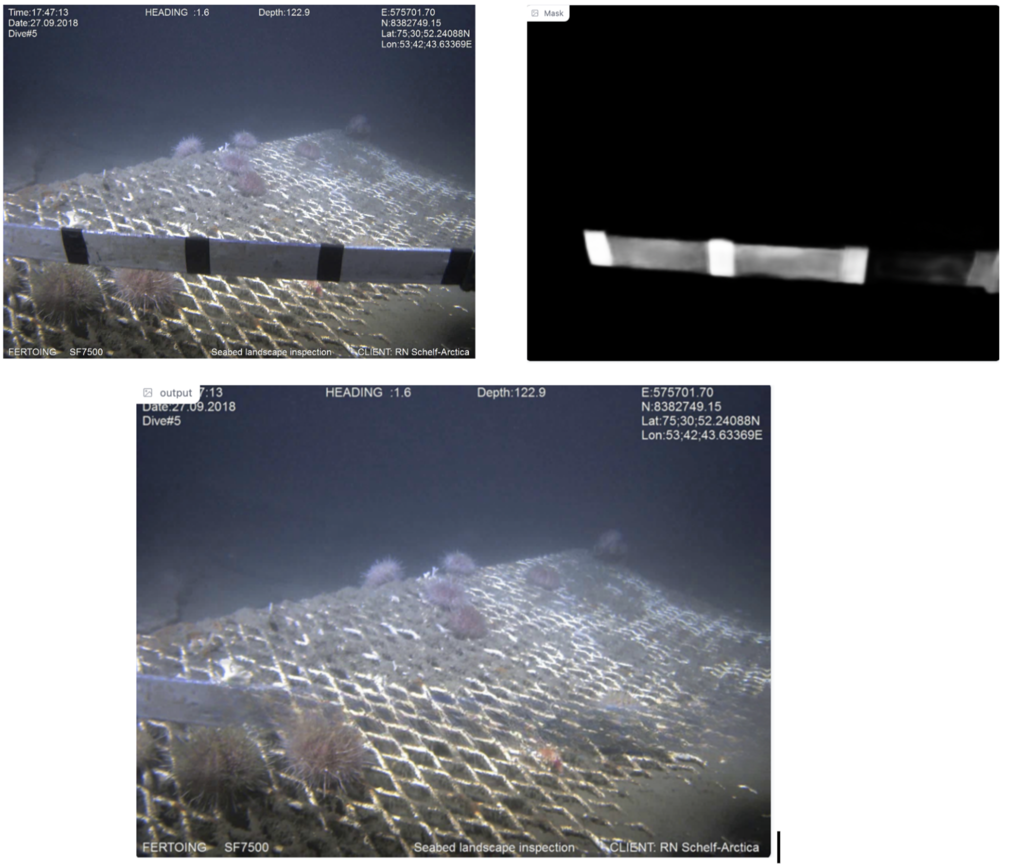

Figure 1: LaMa results: Resolution-robust Large Mask Inpainting with Fourier Convolutions. The first image is the input data, with an object that makes it difficult to solve the problem of segmentation and classification; the second image is an auto-generated mask of the main object in the image; the third image is the result of solving the fragment replacement problem.

3. Replacing Image Fragments

Replacing objects in a video recording, which aims to fill in missing areas of the video, remains a challenging task due to the inability to maintain proper spatial and temporal consistency of video content.

At the moment, the modern fragment inpainting system Lama [4] is one of the three state-of-the-art solutions in the Image Inpainting on Places2 benchmark [5] and others. At the same time, for the inference of this model, there is no need to create an image mask on its own, since the model is able to select the main objects in the image on its own (an example of such a baseline is Fig. 1).

This system is easily adaptable to the task of inpainting high-resolution images underwater, since the model is trained with an emphasis on the ability to work with large masks (a new method is presented in [4]).

4. Classification

Based on the study's results, we have formed a database of objects from the bottom of the White Sea, where research expeditions settle annually for geo- and bio-prospecting. Nevertheless, to train a neural network, it is necessary to have a massive and diverse dataset [6], so some of the data were aggregated from open data sources. Also, at the bottom, there are a large number of specific classes of objects of human activity, such as barrels, bottles, iron cans, and garbage, which are assigned to a separate class. Our database contains 36 classes, 6 of them are generalised concepts such as fish, garbage, and man. For better recognition of classes, aggregated open-source data [7], and data from the White Sea are mixed in a proportion of 60 to 40. As an architectural solution, we used ResNet [8] pre-trained on 1000 classes and re-trained on the classes of interest. As a result of finetuning, this model achieved an accuracy of 95 percent on the validation set.

5. Segmentation

To solve the segmentation problem, DoG-BConvLSTM - a model that solves the few-shot learning (FSL) segmentation problem [9] was used (one of the three state of art solutions with Mean IoU = 83.36). Since, in our case, it would take a very long time to form a dataset for segmentation [10]. The task of few-shot learning segmentation is the ability of the model to segment an object whose class was not previously known, according to several notable examples (fig. 2).

To test the solution to the FSL problem of segmentation of various marine life, a sample of 6 images for each species was created. For this dataset, the original image was resized to 224 X 224 pixels, and a black and white image mask of the same size was created. To solve the FSL segmentation problem, a 6-shot model trained on the FSS-1000 dataset was used [11]. The easy expandability of the FSS-1000 can also be attributed to the positive characteristics of this approach.

The input of the neural network is a support set - a set of 6 pairs of an object image and a black-and-white mask, as well as a frame of the video sequence. It should be noted that before solving the segmentation problem, each frame is divided into 12 parts since the objects of interest are often relatively small. After predicting the model for each of the parts of the frame, the image is merged back.

6. 3D reconstruction

For a more accurate analysis of the biotope, there is a need to recreate a 3D model of the bottom. However, in contrast to the standard formulation of the problem, where a well-lit stationary object is reconstructed, around which the camera flies [12]. In the case of underwater reconstruction, the task becomes much more complicated.

To get the point cloud [13], we used the hierarchical localization [14] toolbox framework. The main stages of which are:

1. feature extraction (SuperPoint: Self-Supervised Interest Point Detection and Description was used)

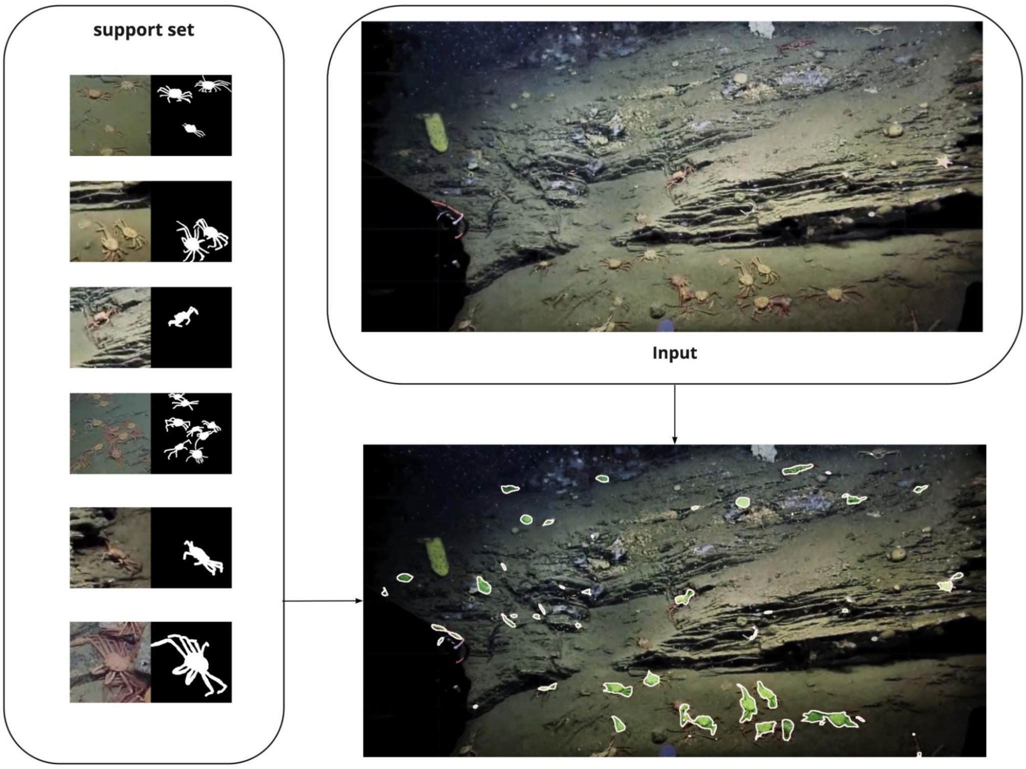

Figure 2: Scheme of work when solving the few-shot learning segmentation problem. Required elements: support set - 6 pairs of labelled images with masks; input image. The result of the work of the neural network is the prediction of areas with an object inside.

2. feature matching (SuperGlue: matching the points found in the previous step using a graph neural network [15])

3. Point Cloud Triangulation (COLMAP: Classical Reconstruction Methods).

To obtain a 3D model, it is necessary to preprocess the generated point cloud (fig. 3). Using the open3d library [16]:

1. remove all selections (points that are farther from each other than the average distance between neighbouring points in the point cloud)

2. change the orientation of the normals (using a minimum spanning tree)

3. Poisson Surface Reconstruction [17]

The result of the presented algorithm is a 3D mesh of the underwater bottom (fig. 4).

Nevertheless, it is worth noting that a good reconstruction detail for large objects (for example, a stone) is not possible, but a detailed reconstruction of the biotope.

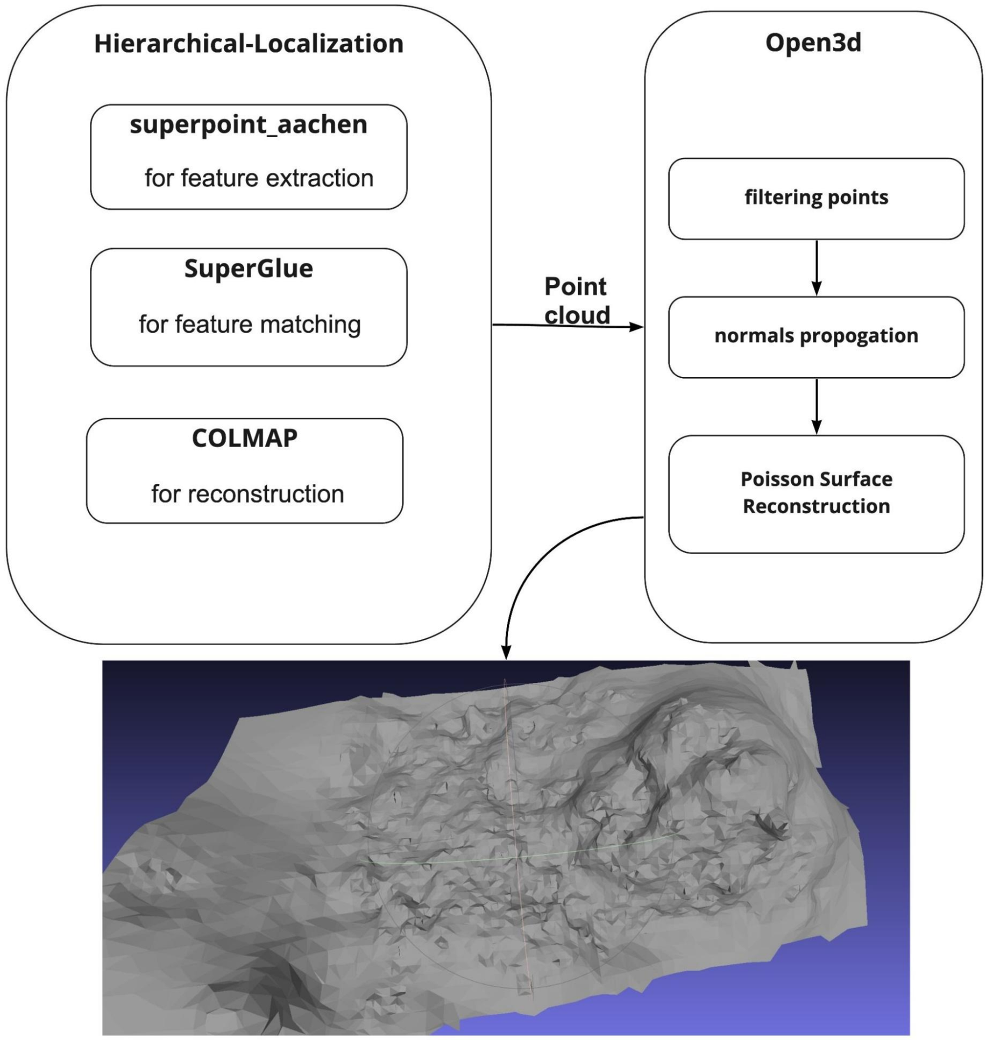

Figure 3: Scheme with key elements of reconstruction to obtain a 3D model of the seabed. The first stage of Hierarchical-Localization are used: SuperPoint, SuperGlue and COLMAP. The result of the work is a point cloud. In the next stage, using the Open3D library, the point cloud is reconstructed into a mesh.

7. Localization

A separate stage is the localisation of objects of interest (fig. 5), which consists in:

1. Objects are segmented on the frame,
2. for each closed contour with an object larger than the given value:
    (a) find the minimum rectangle in which the contour can be inscribed
    (b) cut rectangle out of frame

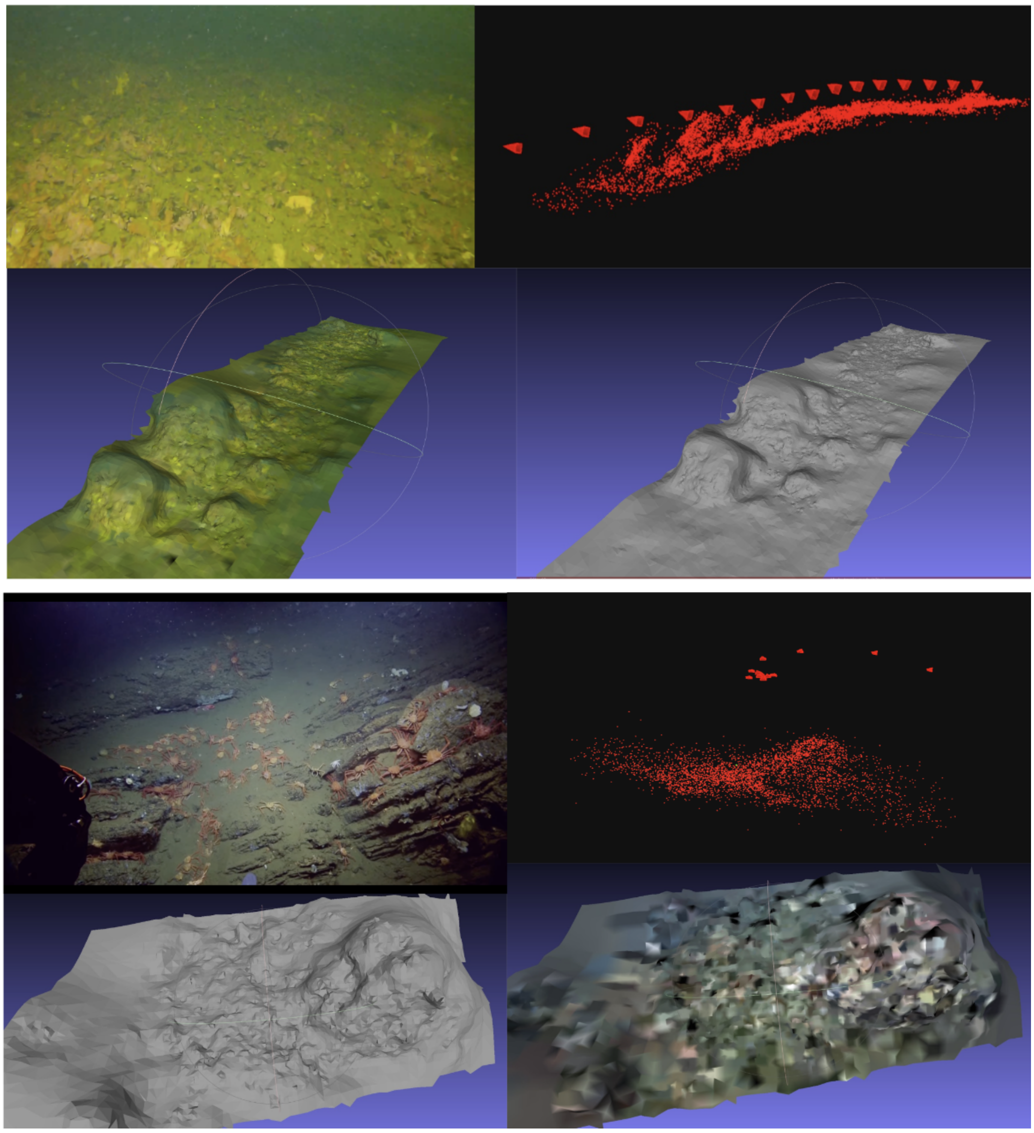

Figure 4: example of 3D reconstruction. Frame from video sequence; Resulting point cloud; Generated mesh; Mesh where colours are stored in points.

The received frame with the object of interest can be localised in the point cloud using hierarchical localisation. The localised points belonging to the object form a surface, the area of which can be calculated and used to estimate the volume of the biotope.

8. Image processing

Part of the solution we propose is presented as a web application, where the launch of a neural network is possible in a browser from any device. The main functionality is to download a video recording, preprocess it, inference a neural network that solves the classification problem, and generate reports in Excel format.

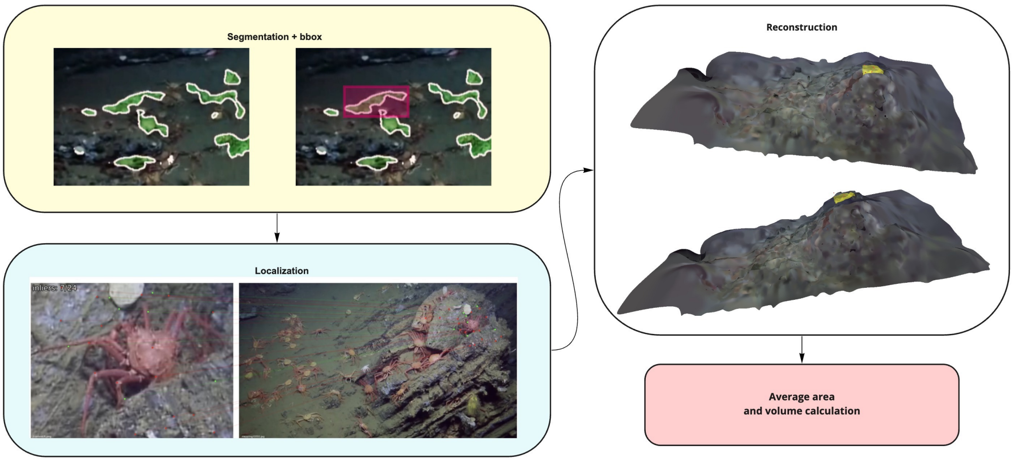

Figure 5: Scheme of the work of estimating the average area and average volume. After segmentation, each contour goes through a localisation stage, according to the results of which the points belonging to the contour are segmented into a 3D model. The segmented area and volume can be roughly calculated.

After loading a video file, a mandatory step in preprocessing the video sequence is splitting it into frames. An additional step is applied for a more accurate solution to the classification problem - splitting each frame into smaller fragments. Which, in turn, are placed in a separate folder. The following critical stage of preprocessing is the filtering of input frames. With the help of frame blur estimation, it is possible to eliminate useless images in terms of solving segmentation and classification problems. Blur evaluation steps:

1. one channel of the image is highlighted (presumably in grayscale);
2. Collapses it with a 3 x 3 core;
3. variance is considered.

If the variance is below the threshold, the image is considered blurry; otherwise, the image will be used in further analysis. The Laplacian obtained from the blur estimation process highlights areas of an image containing rapid changes in intensity and is often used for edge detection. It is assumed that a typical image in focus is characterised by high dispersion.

There are currently two main approaches:

- Using the OpenCV computer vision library, the main one is using the GStreamer framework or packages based on OpenCV.
- Using the FFmpeg library set.

Based on the results of the analysis of existing video sequence decomposition solutions, several advantages of the second approach were revealed, namely: ease of integration with Python, the possibility of using a GPU, the speed of video sequence decomposition, and the presence of additional functionality for splitting a frame into parts (fig. 6). A set of FFmpeg libraries was used

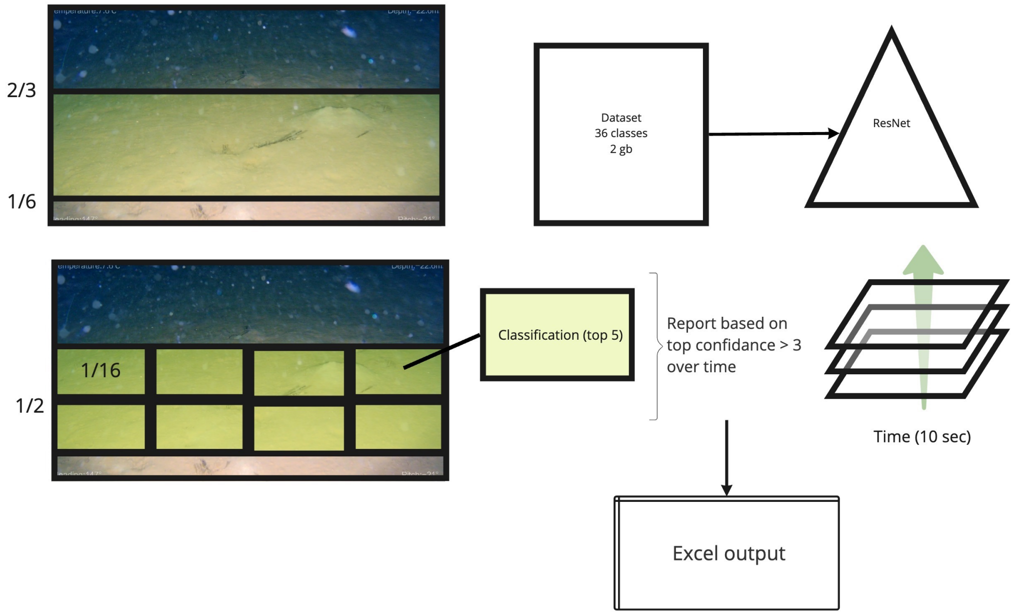

Figure 6: Splitting the frame into parts for further analysis. After selecting the area of interest (bottom surface), the work is divided into parts. A neural network is applied to each part to solve the classification problem. The five most likely outcomes for frames from a 10-second interval are placed in a table.

to speed up the work of the solution, which support the possibility of hardware acceleration of the decomposition process.

To interact with the neural network in the browser [18], our trained model was converted to the Onnx format and imported into the web application using the onnxrunetime library. Based on the results of the model prediction, a report is generated. It is an Excel document with a table where each time interval of the video recording is assigned the number of recognised objects of each class.

Mandatory input data in our system are video files (fig. 7). However, data such as coordinates located on the frame, masks for inpainting interfering objects, and a dataset from known object classes. At the preprocessing stage, the video sequence is divided into separate frames, where each frame goes through the stage of filtering and, if necessary, replaces large objects. The Tesseract OCR library [19] is used to extract geolocation (latitude and longitude) [20]. After going through the preprocessing stage for a sequence of frames, key points are extracted from the video. It is also possible to solve the problem of calculating the surface coverage and counting the number of objects. In addition, according to the sequence of frames and key points, a point cloud is aggregated from which a 3D environment model is reconstructed. The correlation of points in the point cloud and the segmented area is evaluated to generate a report and calculate the average volume of objects in the video. Geolocation and average volume can then be used to generate a report on the study.

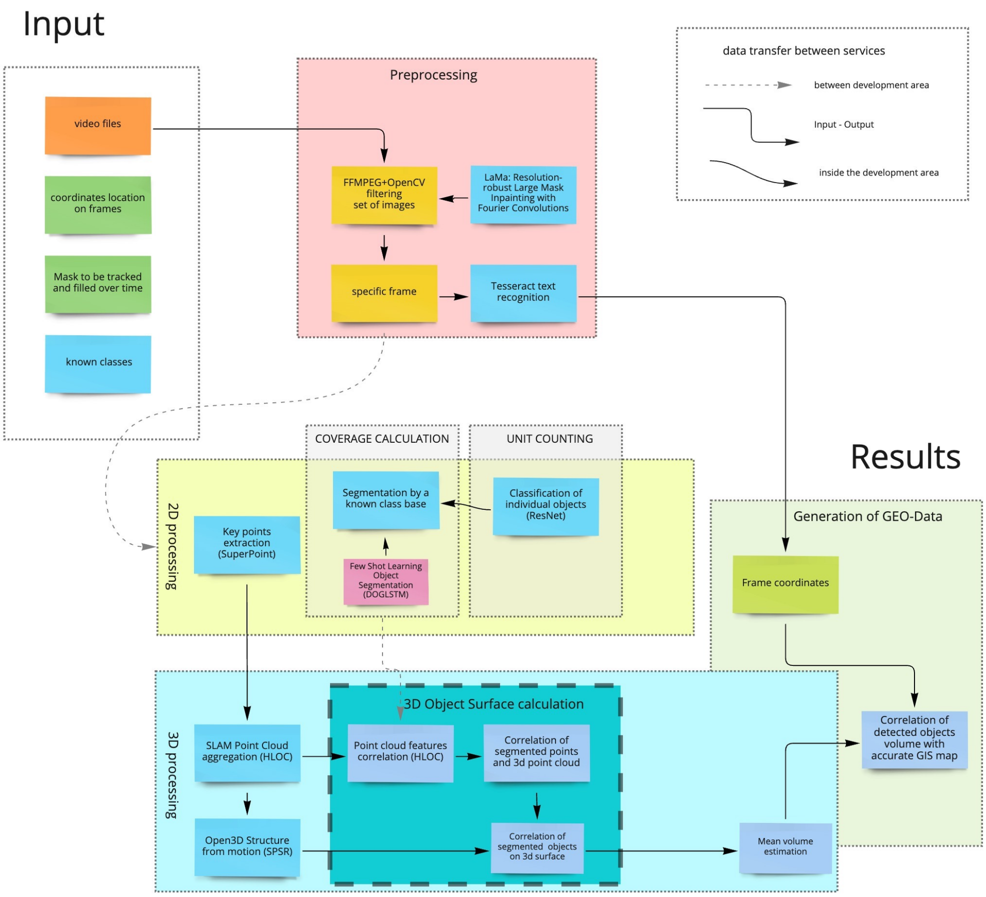
Input
video files
coordinates location on frames
Mask to be tracked and filled over time
known classes
Preprocessing
FFMPEG+OpenCV filtering set of images
LaMa: Resolution-robust Large Mask Inpainting with Fourier Convolutions
specific frame
Tesseract text recognition
data transfer between services
between development area
Input - Output
inside the development area
COVERAGE CALCULATION
UNIT COUNTING
2D processing
Key points extraction (SuperPoint)
Segmentation by a known class base
Classification of individual objects (ResNet)
Few Shot Learning Object Segmentation (DOGLSTM)
Results
Generation of GEO-Data
Frame coordinates
3D processing
SLAM Point Cloud aggregation (HLOC)
3D Object Surface calculation
Point cloud features correlation (HLOC)
Correlation of segmented points and 3d point cloud
Open3D Structure from motion (SPSR)
Correlation of segmented objects on 3d surface
Mean volume estimation
Correlation of detected objects volume with accurate GIS map

Figure 7: Diagram shows a complete pipeline for analysing the underwater environment from video data.

9. Conclusion

Construction, fishing, and mining are just a few practical issues that might be addressed by studying undersea life. A thorough examination carried out by human professionals in the field of undersea life shows how this duty is conducted on an industrial scale. In this study, we demonstrated a technique that we developed that drastically cut down on time needed for underwater video analysis. Our system transfers the laborious work of video inspection to AI, enabling a shortcut that limits professionals from checking all of the results' accuracy. To accomplish this, we have created an observation pipeline that divides the video into frames, evaluates each frame's level of noise and blurriness, makes corrections by increasing resolution, counts the number of animals in each frame, creates a report on the video's content, and plots the data for the biotope on a map. As a result, it takes much less time to analyse underwater video data in comparison with approach wich based on human abilities.

We also took into account the task of calculating biotope mass. To do so, we corresponded the output of a few-shot learning segmentation model to point-cloud data. As a result, we could determine the biotope's surface covering area and estimate its volume. The detailed mapping and surveillance of a region can benefit from such estimation.

Thus, employing autonomous processing of underwater video data from a single camera, we offer a system that enables detailed underwater biotope mapping in this work. To do this, a group of cooperative deep neural network systems are combined into a single pipeline.

References

[1] Ustalov D. Challenges in Data Production for AI with Human-in-the-Loop //Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining. – 2022. – . 1651-1652.

[2] Iakushkin O. et al. Automated Marking of Underwater Animals Using a Cascade of Neural Networks //International Conference on Computational Science and Its Applications. – Springer, Cham, 2021. – P. 460-470.

[3] Zhang S., Zhen A., Stevenson R. L. Deep motion blur removal using noisy/blurry image pairs //Journal of Electronic Imaging. – 2021. – . 30. – №. 3. – . 033022.

[4] Suvorov R. et al. Resolution-robust large mask inpainting with Fourier convolutions //Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. – 2022. – . 2149-2159.

[5] Zhou, Bolei, et al. "Places: A 10 million image database for scene recognition." IEEE transactions on pattern analysis and machine intelligence 40.6 (2017): 1452-1464.

[6] Villon S. et al. Automatic underwater fish species classification with limited data using few-shot learning //Ecological Informatics. – 2021. – . 63. – . 101320.

[7] Nair R. S. et al. Image mining applications for underwater environment management-A review and research agenda //International Journal of Information Management Data Insights. – 2021. – . 1. – №. 2. – . 100023.

[8] Mathur M., Goel N. FishResNet: Automatic Fish Classification Approach in Underwater Scenario //SN Computer Science. – 2021. – . 2. – №. 4. – . 1-12.

[9] Azad, R., Fayjie, A. R., Kauffmann, C., Ben Ayed, I., Pedersoli, M., Dolz, J. (2021). On the texture bias for few-shot cnn segmentation. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (pp. 2674-2683).

[10] Li L. et al. Marine animal segmentation //IEEE Transactions on Circuits and Systems for Video Technology. – 2021. – . 32. – №. 4. – . 2303-2314.

[11] Li X. et al. Fss-1000: A 1000-class dataset for few-shot segmentation //Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. – 2020. – P. 2869-2878.

[12] Fu K. et al. Single image 3D object reconstruction based on deep learning: A review //Multimedia Tools and Applications. – 2021. – . 80. – №. 1. – . 463-498.

[13] Ma B. et al. Surface reconstruction from point clouds by learning predictive context priors //Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. – 2022. – . 6326-6337.

[14] Sarlin P. E. et al. From coarse to fine: Robust hierarchical localisation at large scale //Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. – 2019. – . 12716-12725.

[15] Sarlin P. E. et al. Superglue: Learning feature matching with neural graph networks //Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. – 2020. – . 4938-4947.

[16] Zhou Q. Y., Park J., Koltun V. Open3D: A modern library for 3D data processing //arXiv preprint arXiv:1801.09847. – 2018.

[17] Xu Z. et al. Robust resistance to noise and outliers: Screened Poisson Surface Reconstruction using adaptive kernel density estimation //Computers Graphics. – 2021. – . 97. – . 19-27.

[18] Wang Z. et al. Wearmask: Fast in-browser face mask detection with serverless edge computing for covid-19 //arXiv preprint arXiv:2101.00784. – 2021.

[19] Hegghammer T. OCR with Tesseract, Amazon Textract, and Google Document AI: a benchmarking experiment //Journal of Computational Social Science. – 2022. – . 5. – №. 1. – . 861-882.

[20] Revathi A. S., Modi N. A. Comparative Analysis of Text Extraction from Color Images using Tesseract and OpenCV //2021 8th International Conference on Computing for Sustainable Global Development (INDIACom). – IEEE, 2021. – . 931-936.