МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное
образовательное учреждение высшего образования
«Национальный исследовательский университет ИТМО»
(Университет ИТМО)

Международный научно-образовательный центр
Физики наноструктур

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
GRADUATION THESIS

по теме:

ИССЛЕДОВАНИЕ АЛГОРИТМОВ КОМПЬЮТЕРНОГО ЗРЕНИЯ ДЛЯ ОТСЛЕЖИВАНИЯ ПОДВОДНЫХ ОБЪЕКТОВ

Студент:
Группа № R34372 *Нгуен Тоан*

Руководитель:
доцент, кандидат технических наук *Шаветов Сергей Васильевич*

Санкт-Петербург 2025

СОДЕРЖАНИЕ

INTRODUCTION

Computer vision is one of the most rapidly developing fields in modern artificial intelligence, with applications spanning a wide range of tasks, from autonomous driving to medical diagnostics and industrial automation systems. One of the critical and complex challenges researchers face is object tracking in underwater environments. Underwater conditions are characterized by complex optical distortions, low image contrast, uneven lighting, and various noise effects, which significantly complicate the process of video data processing and analysis.

The relevance of this topic is driven by the need to create reliable monitoring systems and manage underwater robotic complexes, as well as to improve methods for marine security and scientific research of ocean resources. The application of modern machine learning and neural network techniques to solve the problem of object tracking in underwater environments allows for enhanced accuracy and robustness of algorithms, which opens up new opportunities for the practical implementation of developed solutions.

The goal of this work is to investigate and develop computer vision algorithms capable of efficiently tracking underwater objects under significant visual distortions. To achieve this goal, several tasks need to be accomplished: conducting an analytical review of existing methods and algorithms used for tracking objects in complex visual conditions; preparing and analyzing the UOT100 dataset, which contains over 100 video sequences under various underwater conditions; evaluating the effectiveness of classical machine learning methods compared to modern neural network approaches, including algorithms based on correlation filters and Siamese networks; and developing software in Python using libraries such as OpenCV, PyTorch, and TensorFlow to implement and compare the selected algorithms.

The structure of the work includes an introduction that justifies the relevance of the topic and sets the research objectives, an overview of existing solutions, a theoretical part dedicated to studying the principles of computer vision algorithms, a description of the experimental setup and analysis conducted, as well as conclusions and recommendations for further development of the topic. The research carried out will not only identify the strengths and weaknesses of existing

approaches but also propose directions for their improvement in the context of underwater applications.

Thus, the work aims to address practical problems related to improving the efficiency of object tracking systems in extreme conditions, which is of great significance for the development of autonomous underwater technologies and the implementation of new solutions in marine safety and monitoring.

1 PROBLEM STATEMENT

Tracking underwater objects, which refers to the continuous and accurate estimation of a target's position within a video sequence captured underwater, is a critical task with significant implications for marine applications. These applications span across environmental monitoring, resource development, and ecological protection [1], where precise tracking can aid in assessing marine ecosystems, monitoring aquatic life, and ensuring the sustainability of ocean resources. However, the unique characteristics of the underwater environment introduce substantial challenges that complicate the task of object tracking in this setting.

One of the foremost challenges is the issue of **water turbidity and blurring effects**. Underwater optical images are frequently affected by turbidity, which results in significant blurring and fog. This occurs due to the absorption and scattering of light by seawater and various suspended particles, including minerals, salt, sand, and plankton [2][3][4]. The turbid nature of water severely limits visibility, introducing haze that further complicates the clarity of captured images. Additionally, turbulence in the water flow can cause motion blur, which further disrupts the tracking process and reduces the precision of object localization [4].

Another obstacle faced in underwater tracking is **color attenuation and light scattering**. As light travels through water, different wavelengths are absorbed at varying rates, with red light being absorbed more rapidly than other wavelengths [3]. This phenomenon leads to the attenuation of color, which distorts the natural hues of objects and affects the overall image quality. The scattering of light by suspended particles further diminishes light intensity and alters its direction, resulting in color distortion that can make it even more difficult to distinguish between different objects. This attenuation of color and reduction in light intensity contribute to the overall decolorization of underwater scenes, posing a significant challenge for accurate object tracking [3][4][5].

In addition to these optical challenges, **low-contrast targets and occlusion** are persistent problems in underwater environments. Low visibility, low contrast, and low light intensity are common characteristics of underwater images, making it particularly difficult to discern objects and extract meaningful features for tracking [2][4]. Moreover, many marine species exhibit camouflage, blending

seamlessly with their surroundings, which complicates the process of segmentation and accurate identification [3]. Furthermore, occlusion, where one object partially or fully obscures another, presents a substantial problem. This can significantly hinder the ability to maintain a consistent track of the target, especially in dynamic environments where objects frequently move in and out of view [2][3][6].

The underwater environment is also characterized by **dynamic backgrounds with similar-looking objects**, adding another layer of complexity. Marine organisms exhibit a wide range of sizes and shapes, often appearing in dense clusters or schools, such as schools of fish. The presence of these similar-looking objects in the background can lead to confusion, as tracking algorithms may mistakenly identify these objects as the target, causing errors and drift in the tracking process [7]. The complexity and dynamic nature of the underwater scene further exacerbate the challenges faced by tracking algorithms.

Traditional object tracking methods, primarily designed and optimized for open-air environments, struggle to perform effectively in underwater settings [3]. These conventional methods are not robust enough to handle the significant domain shifts introduced by the optical properties of water, such as color distortion, image blur, and light scattering [3]. Moreover, mainstream tracking models can be computationally expensive, often requiring high computational power that may not be available on resource-constrained underwater edge devices [1]. The inherent challenges of target feature distortion and contrast attenuation in underwater scenes can further undermine the ability of general tracking algorithms to differentiate between the target and its background, reducing their reliability. Therefore, specialized tracking models that take into account the unique characteristics of the underwater environment, along with underwater-specific datasets, are essential for improving the accuracy and reliability of object tracking in such challenging conditions [3].

2 OVERVIEW OF EXISTING SOLUTIONS

2.1 Correlation Filter-Based Tracking Methods

Correlation Filter (CF)-Based Tracking has emerged as a prominent approach in visual object tracking, known for achieving a good balance between tracking accuracy and speed [8]. These methods utilize a dynamic model to track the same target across consecutive video frames [9]. The core idea revolves around learning a correlation filter in the frequency domain to efficiently locate the target in subsequent frames [10].

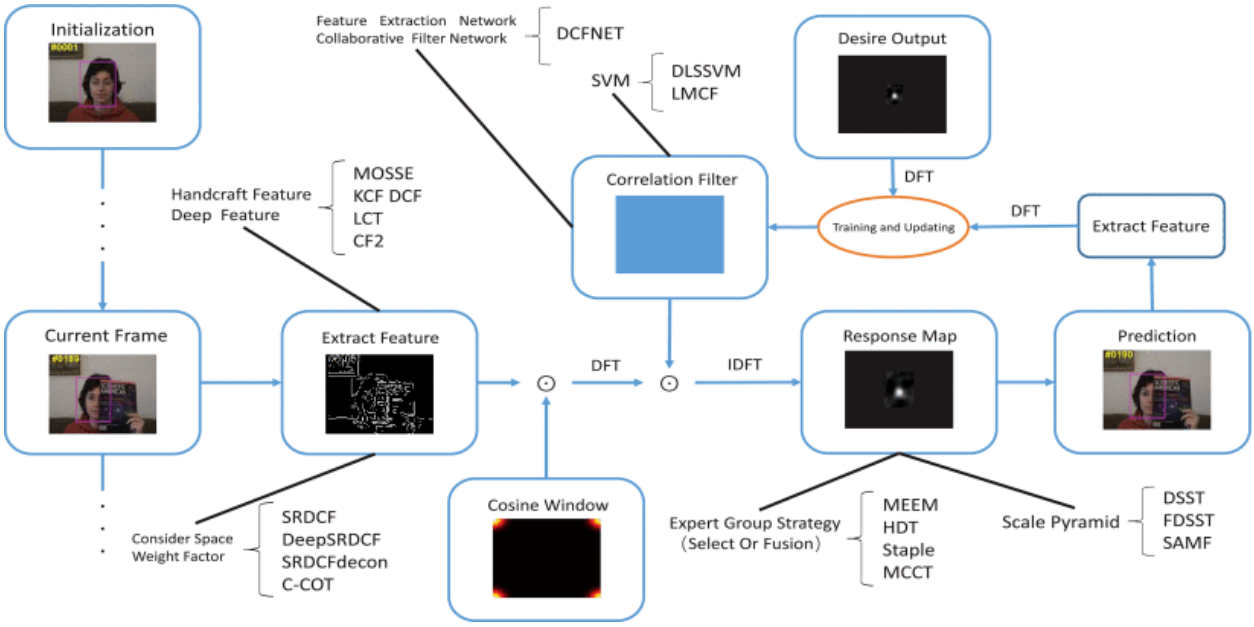

Рисунок 1 — General framework for correlation-filter-based object tracking [9]

The general framework for correlation filter-based object tracking typically involves the following steps [9][10]:

1. **Feature Extraction**: In the initial frame, features are extracted from an image block at the target's location [9] [10]. These features can include handcrafted features like grayscale, Histogram of Oriented Gradients (HOG) [10][9], and color names (CN) [11], or deep features extracted from Convolutional Neural Networks (CNNs) [9][10][11] . A cosine window is often applied to reduce boundary effects [9].
2. **Filter Training:** A correlation filter is learned based on the extracted features and a desired output, often a Gaussian-shaped response map

centered at the target [8][9] . This learning process aims to create a filter that produces a strong correlation response when convolved with the target and a weak response elsewhere [1][9]. Techniques like the Minimum Output Sum of Squared Error (MOSSE) algorithm represent early and straightforward approaches to filter learning [8] [9] [10]. Kernel Correlation Filters (KCF) further enhance this by utilizing a cyclic matrix to construct training samples and employing kernel functions to handle non-linear features [9][10][11]. Dual Correlation Filters (DCF) represent another early development [9].

3. **Target Localization:** In subsequent frames, a search window centered around the previous target location is cropped, and features are extracted [10]. The learned correlation filter is then convolved with these features to generate a response map [9][10]. The location of the maximum response in this map indicates the new position of the target [9][10]. The response map's characteristics, such as its peak value and shape, can also be used for tasks like failure detection [11].
4. **Filter Update:** To adapt to changes in the target's appearance and the environment, the correlation filter is typically updated online with new information from the tracked target in the current frame [1][9][10]. The learning rate of this update can be adaptively determined based on tracking reliability [9].

Correlation filter-based object tracking methods have undergone significant advancements and can be categorized based on several key characteristics [9]. Early approaches utilized categorized features such as grayscale information, exemplified by the Minimum Output Sum of Squared Error (MOSSE) tracker [8][10][11]. Subsequent developments incorporated more sophisticated handcrafted features like Histogram of Oriented Gradients (HOG) and color names [3][7][8][11]. The integration of deep convolutional neural network (CNN) features has further enhanced performance by providing robust and discriminative representations [8][9][11]. Some trackers leverage a combination of handcrafted and deep features to capitalize on their complementary strengths [7][9][11].

Another crucial categorization factor is the consideration of the spatial context through space weight factors. Spatially Regularized Discriminative Correlation Filters (SRDCF) were introduced to mitigate boundary effects

by penalizing filter coefficients outside the target region [9][8][11]. Variants like DeepSRDCF combine SRDCF with deep features [8][9], and CSRDCF incorporates channel and spatial reliability [8][9][10]. Dynamic Saliency-Aware Regularized CF Tracking (DSAR-CF) refines spatial regularization by integrating object saliency information, allowing the weight map to adapt to shape variations [8][9][11]. Scale factors represent another important category. Discriminative Scale Space Tracking (DSST) decouples translation and scale estimation to accurately determine the object's size [7][8][11].

Expert strategies involve utilizing multiple filters or cues and selecting the most reliable one for tracking. Multiexpert Entropy Minimization (MEEM) based on Support Vector Machines (SVM) is an example of this approach [9][12]. Methods employing decision-level fusion also fall under this category, combining the outputs of multiple experts based on different features or filter types [11][3]. Large Margin Object Tracking with Circulant Feature Maps (LMCF) combines correlation filters with structured SVM for robust tracking [8][9].

Recent advancements include the development of datasets tailored for specific domains, such as WebUOT-1M for underwater object tracking [7]. The introduction of Transformer-based architectures in tracking has also shown promising results [7]. Furthermore, the integration of motion estimation and failure correction modules has been explored to enhance tracking robustness in challenging scenarios, such as in satellite videos [11]. The Walsh-Hadamard transform (WHT) has also been investigated for feature extraction in underwater object tracking within a particle filter framework [5]. These categorizations and advancements highlight the continuous efforts to improve the accuracy, robustness, and efficiency of correlation filter-based object tracking algorithms for diverse applications [8][9][10][11].

Correlation filter-based object tracking has evolved significantly with the introduction of various methods that enhance accuracy, robustness, and efficiency. Below is a detailed categorization and description of specific CF-based trackers, highlighting their key features and advancements.

Specific CF-Based Trackers:

– MOSSE (Minimum Output Sum of Squared Error) [11]: An early, fast tracker using grayscale features.

- KCF (Kernelized Correlation Filters) [10]: Utilizes HOG features and kernel functions for improved accuracy and robustness.
- DCF (Dual Correlation Filters) [8]: An early approach based on correlation filters.
- SRDCF (Spatially Regularized Discriminative Correlation Filters) [8]: Addresses boundary effects using spatial regularization.
- DeepSRDCF [10]: Combines deep features with SRDCF
- C-COT (Continuous Convolution Operators for Tracking) [9]: Employs continuous convolution operators and deep features, achieving high accuracy.
- ECO (Efficient Convolution Operators) [8]: An efficient version of C-COT with a focus on speed and reduced overfitting.
- DSST (Discriminative Scale Space Tracking) [8]: Specifically designed for accurate scale estimation.
- SAMF (Scale Adaptive with Multiple Features) [11]: Integrates multiple features and scale estimation.
- MCCTH (Multicue Correlation Tracker-Based Handcrafted Feature) [9]: Uses multiple handcrafted feature experts.
- LMCF (Large Margin With Circulant Feature Maps Tracker) [9]: Combines correlation filters with structured SVM.
- DCFNET (Discriminant Correlation Filters Network) [9]: An end-to-end trainable network for CF tracking.
- STRCF (Spatial-Temporal Regularized Correlation Filter) [11]: Incorporates both temporal and spatial regularization.
- MACF (Motion-Aware Correlation Filter) [11]: Integrates motion estimation for satellite video tracking.
- CSR-DCF (Discriminative Correlation Filter with Channel and Spatial Reliability) [11]: Utilizes channel reliability and spatial confidence.
- BACF (Background-Aware Correlation Filters) [10]: Learns from real negative background examples.
- DSAR-CF (Dynamic Saliency-Aware Regularized CF Tracking) [8]: Uses dynamic saliency-aware regularization.

Correlation filter-based trackers exhibit several notable advantages and limitations. One of the primary strengths of these methods lies in their ability

to achieve a favorable balance between tracking accuracy and computational efficiency. This efficiency is largely attributed to the utilization of the Fast Fourier Transform (FFT), which enables rapid convolution operations in the frequency domain [8]. Additionally, these trackers are capable of learning discriminative filters that effectively distinguish the target from its surroundings [1].

However, traditional correlation filter-based trackers also face several challenges. Trackers relying on shallow features are particularly susceptible to environmental factors such as background clutter, occlusions, variations in illumination, and target deformations [8][9][10]. Furthermore, the inherent circular shift operation in these methods can lead to undesirable boundary effects, although spatial regularization techniques have been developed to address this issue [8]. Another limitation is the sensitivity of these trackers to hyperparameter settings, which often require careful tuning to achieve optimal performance [9].

While correlation filter trackers offer a balance of efficiency and robustness, ongoing research continues to explore avenues for improvement, including the design of more effective filters, the fusion of complementary features, the development of robust scale estimation and occlusion handling techniques, and the adaptation to specific application scenarios [9][10]. The integration of advanced deep learning architectures and attention mechanisms also represents a promising direction for future work [1].

2.2 Deep Learning-Based Tracking Methods

Object tracking based on object detection leverages the advancements in object detection algorithms to identify and track objects across video frames. By detecting objects in each frame and associating them over time, these methods provide a robust framework for tracking. Common deep learning-based object detection algorithms can be categorized into two-stage and single-stage algorithms based on convolutional neural networks (CNNs), as well as transformer-based object detection algorithms [2].

Two-stage object detection algorithms, such as Faster R-CNN, operate by first generating region proposals to identify potential object locations. These proposals are then classified and refined to achieve precise object detection and boundary estimation. This method is particularly effective in scenarios requiring

high accuracy, as it systematically narrows down the search space for object localization [2].

In contrast, **single-stage object detection algorithms**, such as YOLO and SSD, predict object classes and locations in a single step. This streamlined process offers a faster and more efficient approach, albeit sometimes at the cost of accuracy. These models, particularly YOLO, have demonstrated reliability in underwater object detection, where computational efficiency is often critical [13].

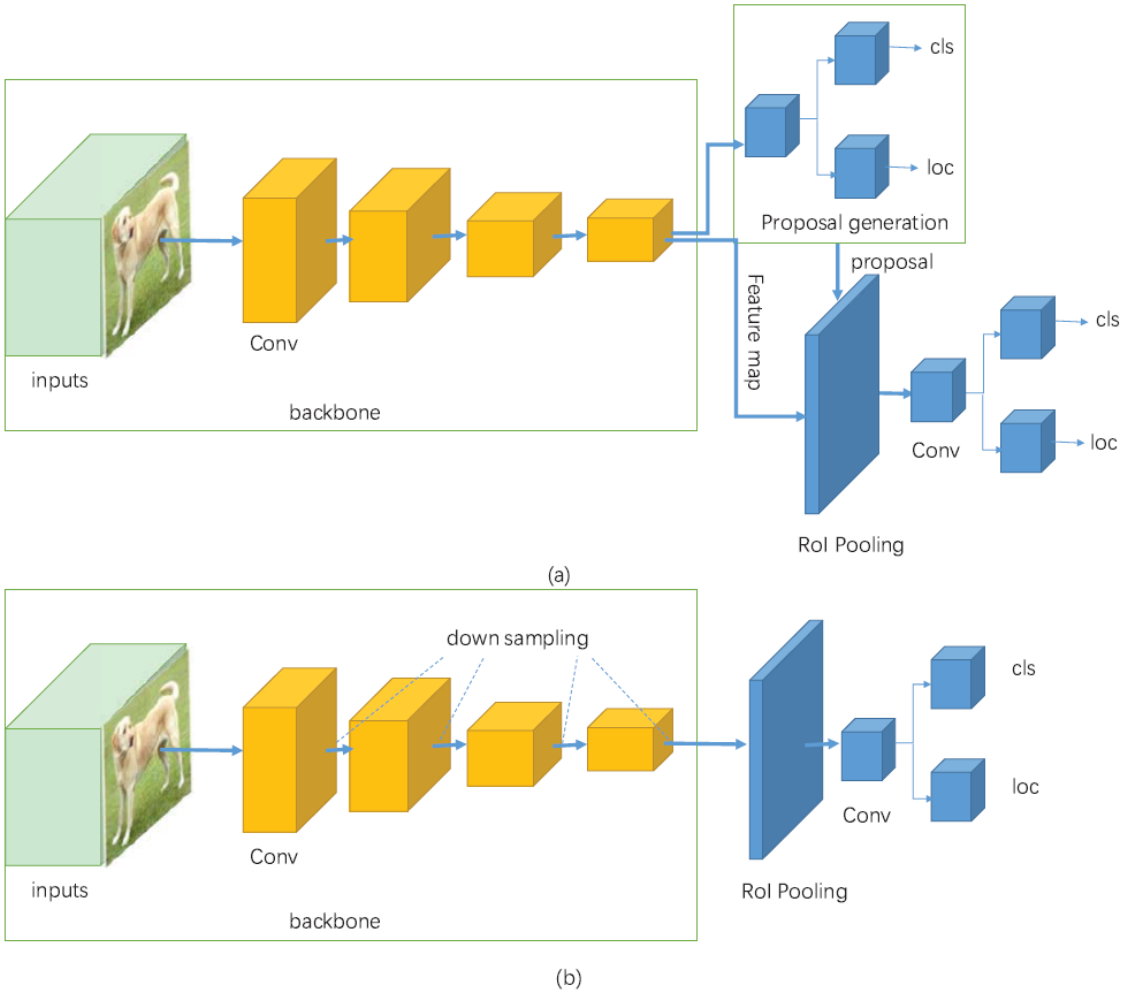

Рисунок 2 — (a) Exhibits the basic architecture of two-stage detectors, which consists of region proposal network to feed region proposals into classifier and regressor. (b) Shows the basic architecture of one-stage detectors, which predicts bounding boxes from input images directly [14].

Additionally, **transformer-based object detection algorithms**, such as DETR and RT-DETR, integrate transformers with convolutional neural networks (CNNs) to directly predict object categories and boundaries. This combination enhances efficiency and accuracy, making them a promising choice for modern object tracking applications [2].

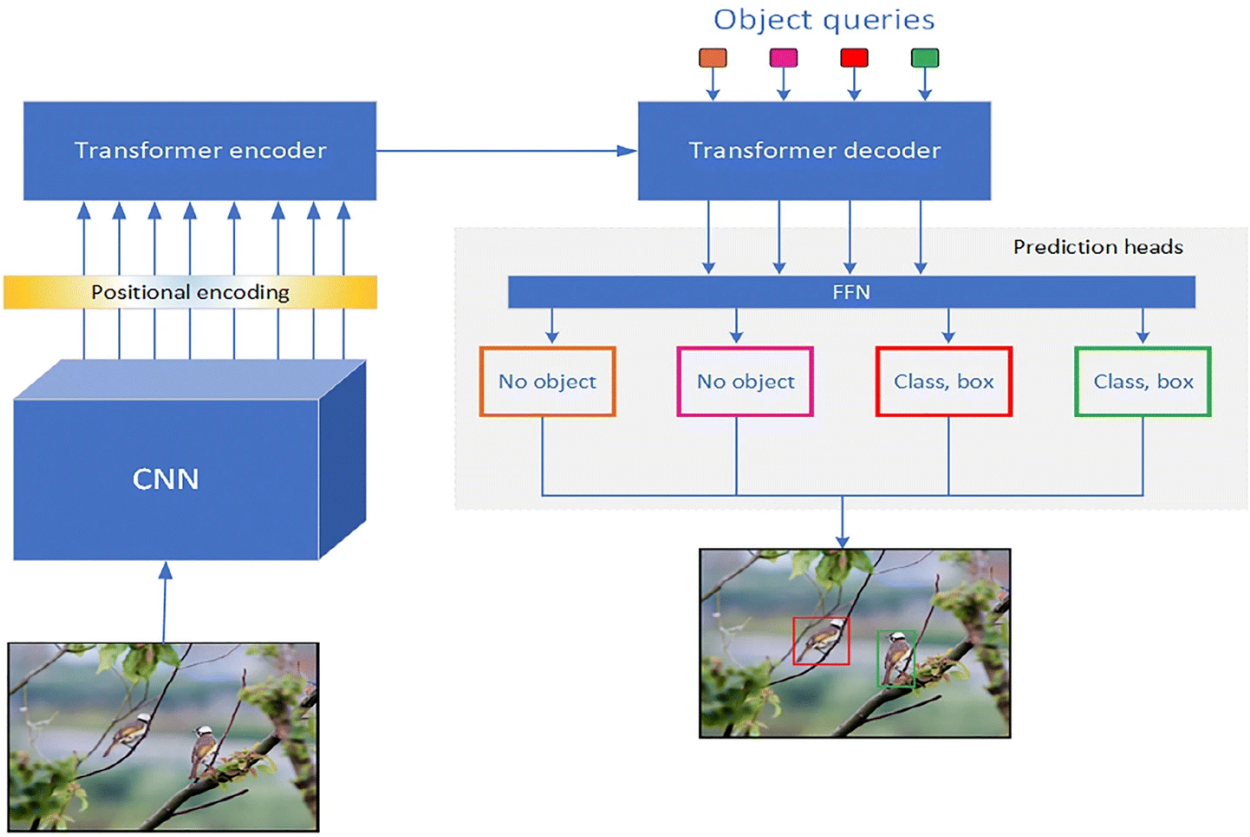

Рисунок 3 — The process of DETR and its structure [15].

Despite demonstrating strong performance in terrestrial environments, these algorithms encounter significant challenges in underwater object detection due to factors such as low light conditions and turbid backgrounds [1][2][6]. Underwater object tracking plays a critical role in applications such as marine resource exploration and military security [1]. Recent advancements in deep learning have garnered considerable attention for their potential in data-driven underwater image enhancement and object recognition [16].

Advanced Deep Learning Techniques Tailored for Underwater Object Tracking (UOT):

– **Knowledge Distillation (KD):** This involves efficiently training a smaller ”student”network by learning from a larger, pre-trained ”teacher”network. This approach has been applied to UOT, where a teacher model trained on massive open-air data can enhance the tracking performance of a student model dedicated to underwater frames [7]. Omni-knowledge distillation combines different types of distillation losses, such as token contrastive representation, similarity matrix, feature embeddings, and response maps distillation losses [7].

- **Motion-Aware Target Prediction (MATP):** To address model drift caused by similar distractors in underwater environments, a training-free MATP module based on Kalman filtering can be used [7]. This involves prediction and correction stages to estimate and refine the target's state. UOSTrack also uses motion-based post-processing to mitigate the influence of similar targets [1][7].
- **Boundary Attention and Sparse Feature Learning:** The DBSF network utilizes a differential boundary attention distribution model to accurately perceive the underwater object edge structure even in noisy and low-resolution images [1]. It then learns to perceive highly discriminative sparse features on the object structure, reducing the computational demands for edge devices [1].
- **Hybrid Training:** Some approaches, like UOSTrack, use hybrid training with both underwater images and open-air sequences to address the sample imbalance problem in underwater datasets [7].
- **Siamese network-based trackers**: Siamese networks employ identical neural networks with shared weights to process the target object's initial appearance (template) and subsequent frames (search region) [3]. By performing a convolutional feature cross-correlation between these two inputs, Siamese networks learn a similarity function to robustly identify the target across varying conditions and frames [3][10][17]. The tracking task is essentially formulated as a similarity-matching function [17].
- **Multi-modal Data:** Leveraging multi-modal data like depth and event information, alongside specialized architectures like multi-scale feature pyramids and transformer-based attention mechanisms, holds significant potential for advancing underwater detection and tracking [3].

Deep learning-based object detection models like YOLO, R-CNN, and SSD are used for underwater tracking [13][3]. Studies have shown YOLO to be reliable for underwater object detection [13]. To enhance temporal stability, YOLO can be merged with SORT [13]. Transformer-based architectures are also showing promising results in UOT [6].

This paradigm involves using object detection models (like YOLOv3) to detect objects in each frame and then linking these detections across frames using tracking algorithms such as Deep SORT, which can be enhanced with LSTM

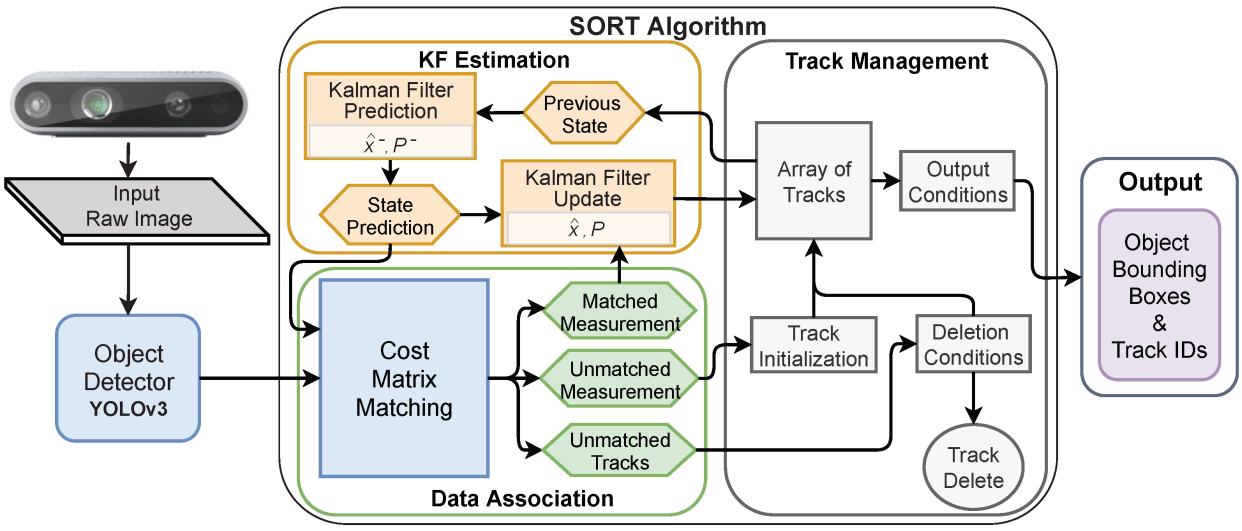

Рисунок 4 — Overview of SORT algorithm based YOLOv3 and Kalman filter [18].

for handling occlusions, as proposed in the HADSYv3 approach [6][3]. This method can be effective in dynamic underwater environments, although its efficacy depends on the quality of the detection algorithm [6].

Several datasets have been established to promote research in UOT, including UOT32, UOT100, UTB180, VMAT, and UVOT400 [7][1][5]. However, many lack training sets or have limitations in size and scenario coverage. The WebUOT-1M dataset has been introduced as a larger benchmark to facilitate the development of more powerful deep UOT algorithms [7]. FishTrack23 is another large-scale dataset specifically for fish tracking [3].

Evaluations on datasets like WebUOT-1M show that Transformer-based trackers (e.g., OKTrack, UOSTrack, All-in-One, GRM, OSTrack) often perform well [7]. UOT-specific trackers like OKTrack and UOSTrack, even with plain ViT backbones, can surpass state-of-the-art open-air trackers, highlighting the domain gap between underwater and open-air environments [7]. Retraining open-air trackers on underwater datasets like WebUOT-1M can effectively reduce this domain gap and improve performance. The DBSF tracker has also demonstrated optimal tracking results on UOT32 and UOT100 benchmarks by focusing on boundary information and sparse confidence features, particularly for edge computing devices [1].

Correlation filter (CF) based tracking offers real-time capabilities, and combining it with deep convolutional features has significantly improved performance [1][9][10][11]. Algorithms like **DeepSRDCF**, which uses CNN features, and **SiamFC**, a fully-convolutional Siamese network, have shown

excellent results [10]. ECO is another efficient correlation filter-based tracker utilizing deep features [10][8][11]. **MACF** is a motion-aware correlation filter algorithm developed for challenging scenarios like satellite videos with small objects and similar distractors, and it has achieved superior accuracy compared to state-of-the-art trackers [11].

Despite the advancements, challenges remain in UOT, including handling occlusion, small or camouflaged objects, low visibility, optimizing computational efficiency for real-time processing on autonomous systems, and mitigating model drift [1][6][7][11]. Future work includes introducing spatial-temporal features and semi-supervised learning to improve tracking in complex underwater scenes [1][6]. Enhancing the detection of small or camouflaged objects, optimizing computational efficiency, and improving real-time processing capabilities are also crucial for deployment in autonomous systems [3]. For correlation filter-based tracking, future research can focus on designing better filters by considering categorized features, space weight factors, scale factors, and expert strategies, as well as combining manual and deep features [9].

2.3 Sensor Fusion-Based Tracking Methods

Sensor fusion-based approaches in underwater environments leverage data from multiple sensors to enhance surveillance and tracking capabilities [19][20][21]. In contrast to traditional methods that rely on single platforms like submarines or frigates, distributed networks of stationary and mobile sensors, such as autonomous underwater vehicles (AUVs), offer advantages in scalability, robustness, and reliability through intelligent networking [19]. **Distributed information fusion (DIFFUSION) strategies**, where local information is shared among sensors, are a key aspect of these intelligent networks [19]. This allows for the combination of data from spatially separated sensors to achieve higher performance [19].

Two primary DIFFUSION schemes are proposed for underwater surveillance: one based on the sharing of local contacts generated by the detection stage, and another based on the sharing of tracks produced by the local tracking stage [19]. In the contact-sharing scheme, contacts are combined at each node using optimal Bayesian tracking based on the random finite set (RFS) formulation [19]. The track-sharing scheme employs a track-to-track (T2T) association and

fusion procedure [19]. A local tracker on each AUV provides a set of tracks and their associated covariance matrices, which are then associated and fused with tracks from other AUVs to obtain more accurate state estimates [19]. Unassociated tracks are treated as originating from a single sensor [19].

The problem of sensor selection within a large network is also critical for optimizing tracking performance under constraints like communication bandwidth [20]. Efficient search techniques, such as convex optimization followed by greedy local search, can be employed to determine near-optimal sensor utilization strategies in real-time for multitarget tracking [20]. Different approaches to sensor selection include ”closest-sensor”strategies, which select sensors nearest to estimated target positions, and more sophisticated methods like ”coarse-step”and ”fine-step”planning that consider tracking performance over time [20]. The posterior Cramér–Rao lower bound (PCRLB) can serve as a basis for network management and as a cost function for optimization [20].

In the context of passive sensor fusion for underwater surveillance, the **Generalized Labeled Multi-Bernoulli (GLMB)** filter can be used for joint filtering of measurements from multiple sensors, enabling multi-target tracking [21]. While detection sets from passive sensors may be staggered in time, the iterated multi-sensor update approach in the GLMB filter has proven sufficiently accurate for simultaneous detections [21]. This Bayesian recursion over time allows for long-term integration of sensor data, even when dealing with the non-linearities inherent in bearings-only tracking, often tackled using sequential Monte Carlo methods [21].

3 THEORETICAL STUDY OF ALGORITHMS

In this study, we employ Visual Object Tracking (VOT) methods to address the problem of underwater object tracking. VOT plays a crucial role in tracking objects across video frames, allowing for continuous localization without prior category knowledge. This approach is essential for handling dynamic underwater environments, where objects may undergo significant appearance changes due to lighting variations, occlusions, and water turbulence.

The field of VOT encompasses a diverse range of algorithmic approaches, broadly categorized into traditional correlation filter-based methods and modern deep learning-based techniques. Correlation filter-based trackers leverage frequency-domain operations for efficient tracking [24; 25], whereas deep learning-based methods exploit the representational power of neural networks to enhance robustness against appearance variations and challenging environmental conditions [26; 27]. This section presents a comprehensive theoretical study of key tracking algorithms, providing insights into their underlying principles and methodologies.

3.1 Minimum Output Sum of Squared Error (MOSSE)

The Minimum Output Sum of Squared Error (MOSSE) tracker, introduced by Bolme et al. [22], represents a pioneering approach in correlation filter-based object tracking. MOSSE employs adaptive correlation filters to efficiently track objects in real-time while maintaining robustness against variations in illumination, occlusions, and background clutter.

Formulation of MOSSE

The fundamental objective of MOSSE is to learn an optimal correlation filter w^* that minimizes the sum of squared errors between the actual output of the convolution operation and the desired response, typically modeled as a Gaussian function centered on the target location. The optimization problem is formulated as follows:

$$\arg\min_{w^*} \sum_i \|f_i \odot w^* - y_i\|^2, \tag{1}$$

where:

- f_i represents the extracted feature map from the i-th input frame,
- w^* is the correlation filter to be learned,
- y_i is the desired response map modeled by a Gaussian function,
- $\odot$ denotes element-wise multiplication.

Solution in the Frequency Domain

Direct computation in the spatial domain is computationally expensive. Leveraging the Convolution Theorem, which states that convolution in the spatial domain is equivalent to element-wise multiplication in the frequency domain, MOSSE reformulates the optimization problem using the Discrete Fourier Transform (DFT):

$$\hat{w}_i = \frac{\sum_i \hat{y}_i \odot \hat{f}_i^*}{\sum_i \hat{f}_i \odot \hat{f}_i^*}, \tag{2}$$

where $\hat{\cdot}$ denotes the Fourier transform, and $(\cdot)^*$ represents the complex conjugate.

Once the optimal filter $\hat{w}_i$ is obtained, object localization in a new frame is determined by computing the response map:

$$\hat{y}_m = \sum_{i=1}^{D} \hat{w}_i \odot \hat{f}_i. \tag{3}$$

Applying the inverse Fourier transform recovers the response map in the spatial domain:

$$y_m = \mathcal{F}^{-1}(\hat{y}_m). \tag{4}$$

The target's new location is estimated as the coordinates of the maximum value in y_m [22].

Online Filter Update Mechanism

To account for appearance variations such as lighting changes and partial occlusions, MOSSE employs an online filter update strategy. The filter numerator A_i and denominator B_i are updated adaptively using a learning rate η:

$$\begin{aligned} A_i &= \eta \hat{y}_i \odot \hat{f}_i^* + (1-\eta) A_{i-1}, \\ B_i &= \eta \hat{f}_i \odot \hat{f}_i^* + (1-\eta) B_{i-1}. \end{aligned} \tag{5}$$

The updated filter for subsequent frames is computed as:

$$\hat{w}_i = \frac{A_i}{B_i}. \tag{6}$$

This adaptive mechanism enables MOSSE to maintain tracking performance despite dynamic changes in the target's appearance [22].

Advantages and Limitations

MOSSE is well-regarded for its computational efficiency, achieving high-speed tracking rates suitable for real-time applications. Key advantages include:

- Robustness to lighting variations and partial occlusions,
- Efficient learning with a low computational footprint,
- Real-time performance with high frame rates.

However, MOSSE has limitations, particularly in handling:

- Scale variations, as it assumes a fixed target size,
- Background clutter in complex tracking environments,
- Non-rigid object deformations.

Conclusion

MOSSE introduced a groundbreaking framework for object tracking using correlation filters, demonstrating real-time performance with adaptive learning. While its limitations have prompted the development of more advanced trackers, such as Kernelized Correlation Filters (KCF) [23], MOSSE remains a foundational method in visual object tracking due to its simplicity and efficiency.

3.2 Kernelized Correlation Filter (KCF)

The Kernelized Correlation Filter (KCF), introduced by Henriques et al. [23], extends the MOSSE framework by incorporating the kernel trick to project image features into a higher-dimensional space. This non-linear transformation improves the discriminative power of the learned filter, enhancing tracking robustness against appearance variations such as scale changes, deformations, and occlusions [24].

Unlike MOSSE, which operates in the linear domain, KCF utilizes a kernel function $k(x, x')$ that implicitly maps the input data into a high-dimensional feature space without explicit computation. This allows for the use of more complex decision boundaries while maintaining computational efficiency through the use of circulant matrices and the Fast Fourier Transform (FFT) [23].

The optimization problem in KCF follows a similar formulation to MOSSE but extends it using a kernel function:

$$\arg\min_{\alpha} \sum_{i} \left\| \sum_{j} \alpha_j k(x_i, x_j) - y_i \right\|^2 + \lambda \|\alpha\|^2, \tag{7}$$

where:

- $k(x_i, x_j)$ is the kernel function measuring similarity between image patches x_i and x_j.
- α represents the dual coefficients of the kernel function.
- λ is a regularization parameter to prevent overfitting.

The solution to the above optimization problem in the Fourier domain is given by:

$$\hat{\alpha} = \frac{\hat{y}}{\hat{k} + \lambda}, \tag{8}$$

where $\hat{k}$ is the Discrete Fourier Transform (DFT) of the kernelized correlation function [23].

The most commonly used kernel in KCF is the Gaussian kernel:

$$k(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2} \right), \tag{9}$$

which ensures a smooth response map and improved tracking stability [24].

Once the optimal filter $\hat{\alpha}$ is learned, object localization in a new frame is determined via element-wise multiplication in the Fourier domain:

$$\hat{y}_m = \hat{\alpha} \odot \hat{k}. \tag{10}$$

Finally, the inverse Fourier transform is applied to recover the response map in the spatial domain:

$$y_m = \mathcal{F}^{-1}(\hat{y}_m). \tag{11}$$

The peak value of y_m represents the object's estimated location in the current frame [23].

Scale Adaptation in KCF

One limitation of the original KCF tracker is its inability to handle scale variations effectively. To address this, an improved version, known as KCF with scale estimation, introduces a multi-scale search mechanism [25]. Instead of relying solely on fixed-size bounding boxes, KCF evaluates different scaled versions of the target appearance and selects the one with the highest response score.

The scale estimation process involves creating a set of scaled versions of the target patch and applying the learned correlation filter to each scale. The optimal scale s^* is selected based on the peak response:

$$s^* = \arg\max_s \mathcal{F}^{-1}(\hat{\alpha} \odot \hat{k}_s), \tag{12}$$

where $\hat{k}_s$ represents the kernelized correlation response at scale s. This scale-adaptive mechanism significantly improves tracking performance in scenarios involving object size variations [26].

Enhancements in Feature Representations

The original KCF tracker operates on raw intensity values, but subsequent implementations have incorporated more advanced feature representations, such as:

- **Histogram of Oriented Gradients (HOG)**: Improves robustness to illumination and texture variations [27].

- **Color Names (CN)**: Enhances performance in scenes with complex backgrounds [28].
- **Deep Convolutional Features**: Extracted from pre-trained CNNs, allowing for superior target representation [29].

By integrating these features, modern KCF variants achieve significantly higher accuracy and robustness in real-world tracking applications.

Computational Efficiency and Practical Considerations

Despite its improvements, KCF remains computationally efficient due to its reliance on FFT and circulant matrix properties. The overall complexity remains $\mathcal{O}(n \log n)$, making it suitable for real-time applications [23]. However, practical considerations such as parameter tuning (e.g., regularization weight λ and kernel bandwidth σ) can impact performance, requiring empirical optimization for different datasets [30].

Summary of KCF Improvements

In summary, KCF extends MOSSE by introducing:

- **Kernel methods** for improved discrimination [23].
- **Scale adaptation** for handling object size variations [25].
- **Advanced feature representations** for better robustness [29].
- **Computational efficiency** suitable for real-time tracking [23].

These enhancements make KCF a widely adopted approach in visual object tracking, balancing accuracy, robustness, and efficiency.

3.3 Generic Object Tracking Using Regression Networks (GOTURN)

GOTURN, proposed by Held et al. [31], is a deep learning-based tracker that employs a regression network for fast and robust visual tracking. Unlike correlation filter-based approaches such as KCF, which rely on frequency domain operations, GOTURN formulates tracking as a bounding box regression problem, leveraging a deep convolutional neural network (CNN) to predict object motion from one frame to the next.

Unlike many conventional trackers that perform online adaptation, GOTURN is trained offline on a large dataset of labeled video sequences and images. This enables the tracker to generalize to novel objects at test time without requiring per-sequence fine-tuning [31]. The model learns a generic mapping between object appearance and motion, making it significantly faster than traditional deep learning-based trackers that involve online learning [31].

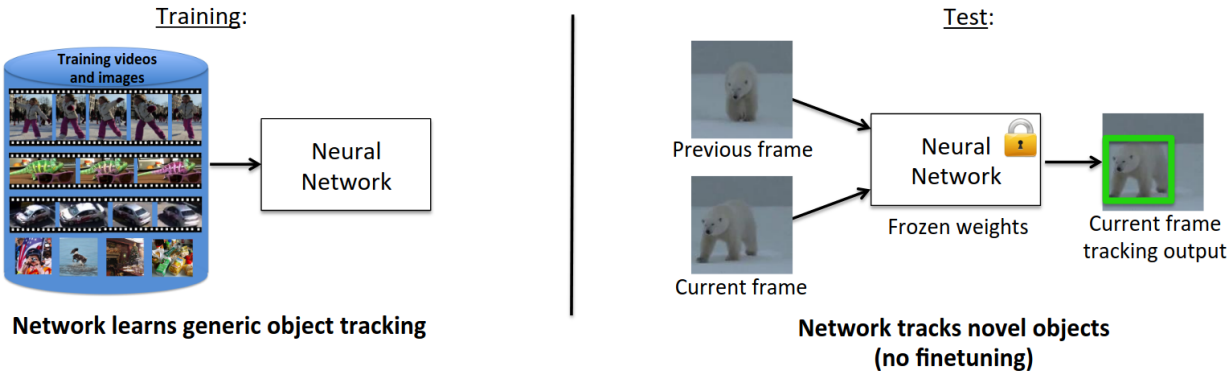

Рисунок 5 — A deep neural network is trained using a combination of labeled videos and images to track generic objects. At test time, the network generalizes to unseen objects without requiring fine-tuning, achieving real-time speeds of 100 FPS [31].

Network Architecture

GOTURN consists of a CNN-based regression model that takes as input two consecutive image patches: the search region from the current frame and the target appearance from the previous frame. The network processes these inputs separately through a series of convolutional layers before merging them into a fully connected layer that outputs the coordinates of the tracked object's bounding box in the current frame [31].

The architecture in Picture 6 follows a two-stream design:

- The first stream extracts features from the target patch in the previous frame.
- The second stream extracts features from the search region in the current frame.
- The feature maps from both streams are concatenated and passed through fully connected layers to predict the new bounding box.

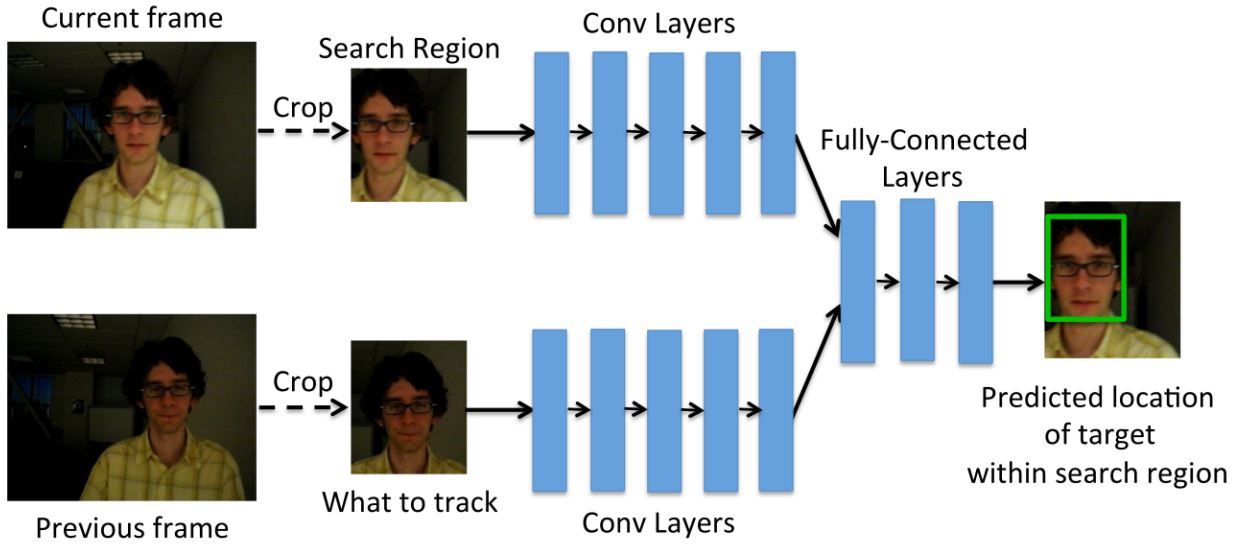

Рисунок 6 — The GOTURN architecture [31].

The convolutional layers are adapted from the CaffeNet architecture [32], which is a variant of AlexNet [33]. The final regression layer outputs the new bounding box coordinates relative to the search region [31].

Loss Function and Training Strategy

GOTURN uses an L1 loss function to minimize the difference between the predicted bounding box and the ground truth:

$$L = \sum_{i=1}^{N} \left| \hat{b}_i - b_i \right|, \tag{13}$$

where b_i represents the ground-truth bounding box coordinates, and $\hat{b}_i$ denotes the predicted bounding box.

The network is trained on a combination of labeled video sequences and still images. The video sequences allow the model to learn temporal object motion, while the still images augment the training data by simulating motion via artificial transformations [31]. The training data is further enhanced using a motion smoothness assumption, where small object movements are favored over large, abrupt changes, improving robustness to motion blur and occlusions.

Tracking Mechanism

During inference, GOTURN follows a simple tracking procedure:

1. The tracker is initialized with a ground-truth bounding box in the first frame.
2. For each subsequent frame, a search region centered around the previous bounding box is extracted.
3. The network predicts the new bounding box coordinates within the search region.
4. The predicted bounding box is used to update the target location.

This approach allows GOTURN to operate without online fine-tuning, enabling real-time tracking at speeds up to 100 FPS on a GPU [31].

Performance and Limitations

GOTURN achieves state-of-the-art performance on the VOT-2014 benchmark [31], outperforming many traditional trackers in terms of accuracy and robustness. However, its reliance on offline training means it does not adapt to target appearance changes during tracking. Additionally, unlike correlation filter-based trackers such as KCF, it lacks an explicit scale estimation mechanism, making it less effective for tracking objects with large-scale variations [26].

Summary of GOTURN Contributions

GOTURN introduces several key advancements in deep learning-based tracking:

- **Offline training:** Unlike traditional trackers that learn online, GOTURN trains a deep network offline, improving speed and generalization [31].
- **Regression-based tracking:** Directly predicts bounding box locations using CNN-based feature extraction and fully connected regression layers [31].
- **Real-time performance:** Achieves up to 100 FPS, significantly faster than previous deep learning-based trackers [31].

Despite its limitations in handling appearance changes and scale variations, GOTURN remains a pioneering approach in deep learning-based object tracking, demonstrating the potential of regression networks for real-time applications.

3.4 Siamese Region Proposal Network (SiamRPN)

Siamese Region Proposal Network (SiamRPN), introduced by Li et al. [34], is an extension of the Siamese network-based tracking framework that incorporates a region proposal network (RPN) for precise and efficient object localization. Unlike GOTURN, which directly regresses bounding box coordinates, SiamRPN applies a matching-based approach where the network learns a similarity function between the target template and candidate regions in the search space.

Network Architecture

SiamRPN follows a Siamese network design, consisting of two identical branches that extract deep feature embeddings from the target template and the search region. The key innovation in SiamRPN is the integration of an RPN module, which enables the network to generate multiple object proposals and select the most confident prediction [34].

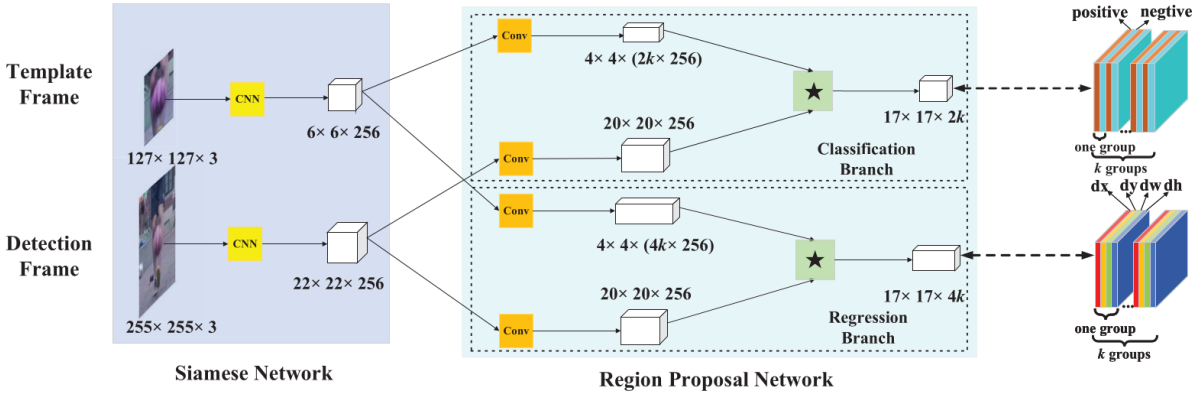

Рисунок 7 — The SiamRPN architecture [34].

The network operates as follows:

- The target template from the first frame is passed through a convolutional backbone (e.g., AlexNet or ResNet).
- The search region from the current frame is processed through the same backbone.
- A cross-correlation operation between the two feature maps is performed to compute response maps, representing object similarity.
- The RPN module generates multiple anchor-based proposals and refines the bounding box predictions.

This pipeline enables SiamRPN to perform both localization and classification simultaneously, improving tracking accuracy compared to standard Siamese trackers such as SiamFC [35].

Loss Function and Training Strategy

SiamRPN optimizes a multi-task loss function, which includes:

$$L = L_{cls} + \lambda L_{reg}, \tag{14}$$

where:

- L_{cls} is the classification loss, typically a cross-entropy loss for distinguishing target vs. background.
- L_{reg} is the bounding box regression loss, commonly a smooth L1 loss for refining object proposals.
- λ is a weighting factor to balance the two objectives.

Training SiamRPN requires large-scale video datasets with ground-truth annotations, such as ImageNet VID and YouTube-BB [34]. The network is pre-trained on static images and fine-tuned on video sequences to learn motion-aware representations.

Performance and Advantages

SiamRPN achieves state-of-the-art results on benchmarks such as OTB-2015 and VOT-2018, surpassing traditional trackers in terms of accuracy and robustness [34]. The main advantages of SiamRPN include:

- High-speed tracking: Runs at over 160 FPS due to efficient network design.
- Robust localization: RPN improves bounding box estimation.
- Scale adaptation: Handles object size variations better than fixed-scale trackers.

These characteristics make SiamRPN a widely used tracker in real-time applications, including autonomous driving and video surveillance.

Comparison with GOTURN

Compared to GOTURN, SiamRPN introduces several key improvements:

- Matching-based tracking (SiamRPN) vs. regression-based tracking (GOTURN).
- Anchor-based bounding box proposals vs. direct bounding box prediction.
- Better scale handling due to multi-scale region proposals.

Overall, SiamRPN represents a significant advancement in deep learning-based tracking, combining the efficiency of Siamese networks with the precision of RPN-based object detection.

Variants of SiamRPN

Following the success of SiamRPN, several improvements have been proposed to enhance its tracking performance, robustness, and efficiency. These variants introduce modifications to the backbone, region proposal strategy, and feature aggregation techniques.

SiamRPN++ [36] improves upon SiamRPN by utilizing deeper backbones such as ResNet and introducing spatial-aware sampling strategies to mitigate the limitations of small receptive fields in shallow networks. The network also applies depth-wise separable convolutions to enhance efficiency, making it more robust to large-scale variations.

SiamBAN (Balanced Anchor-free Network) [**chen2020siamban**] replaces the anchor-based RPN module with an anchor-free mechanism, reducing computational complexity while improving accuracy. Instead of predefined anchor boxes, SiamBAN directly predicts the target's center location and bounding box size, enhancing adaptability to scale variations.

SiamCAR (Classification and Regression) [37] further refines the anchor-free tracking paradigm by employing a dense prediction strategy. The network jointly classifies and regresses bounding boxes at each location in the feature map, improving tracking precision while maintaining high speed.

LightTrack [38] emphasizes lightweight and efficient tracking by employing neural architecture search (NAS) to design a compact and optimized network. By balancing accuracy and computational efficiency, LightTrack

achieves competitive performance while being well-suited for deployment on resource-constrained devices such as mobile and edge platforms.

SiamGAT [39] introduces graph attention networks (GATs) into the Siamese tracking framework, allowing the model to better capture spatial relationships between objects and background features. This enhances tracking performance in cluttered scenes and improves robustness to occlusion.

3.5 Transformer Tracker

Transformer-based models have recently emerged as powerful alternatives to conventional correlation-based tracking frameworks. Unlike traditional approaches such as MOSSE [22] and KCF [23], which rely on handcrafted features and linear correlation, or deep-learning-based methods like SiamRPN [34] and GOTURN [31], which incorporate convolutional networks for feature extraction, Transformer-based tracking frameworks leverage self-attention mechanisms to enhance feature fusion and robustness.

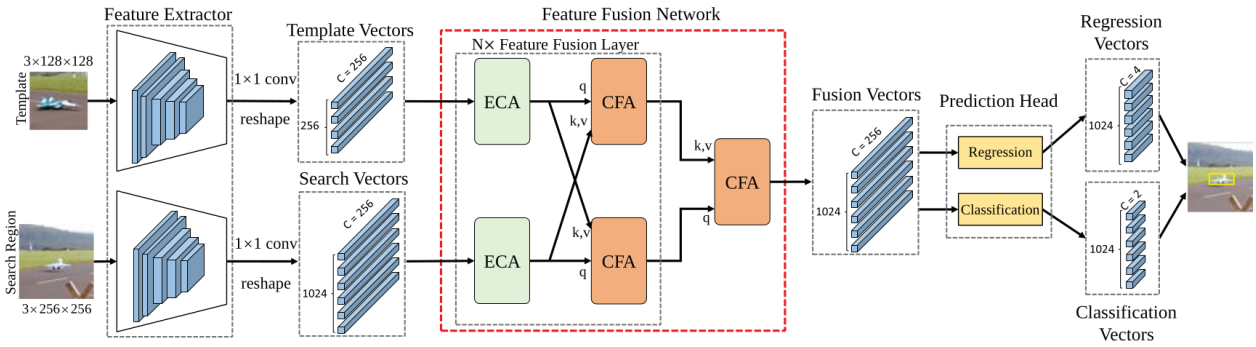

Рисунок 8 — Architecture of Transformer tracking framework [40].

Network Architecture

The Transformer architecture, originally proposed for natural language processing [41], has been successfully adapted for object tracking. The core idea behind Transformer tracking is the replacement of the conventional cross-correlation operation with an attention-based feature fusion mechanism. One notable example is TransT [40], which introduces an attention-driven network consisting of an ego-context augment module (ECA) and a cross-feature augment module (CFA). These components effectively enhance global feature interactions between the target template and the search region.

Feature Extraction: Transformer-based trackers typically utilize a Siamese backbone for feature extraction. Given an input template $z \in \mathbb{R}^{3\times H_z\times W_z}$ and a search region $x \in \mathbb{R}^{3\times H_x\times W_x}$, a shared feature extractor (often based on ResNet-50 [42]) processes both inputs to generate feature maps f_z and f_x, which are then forwarded to the feature fusion network.

Attention-Based Feature Fusion: Unlike previous methods that apply depthwise cross-correlation for similarity computation, Transformer trackers employ self-attention and cross-attention modules:

– **Ego-Context Augment (ECA):** Enhances local features by capturing long-range dependencies within the template and search region through multi-head self-attention.
– **Cross-Feature Augment (CFA):** Establishes interdependencies between template and search region features, allowing adaptive fusion without explicit correlation operations.

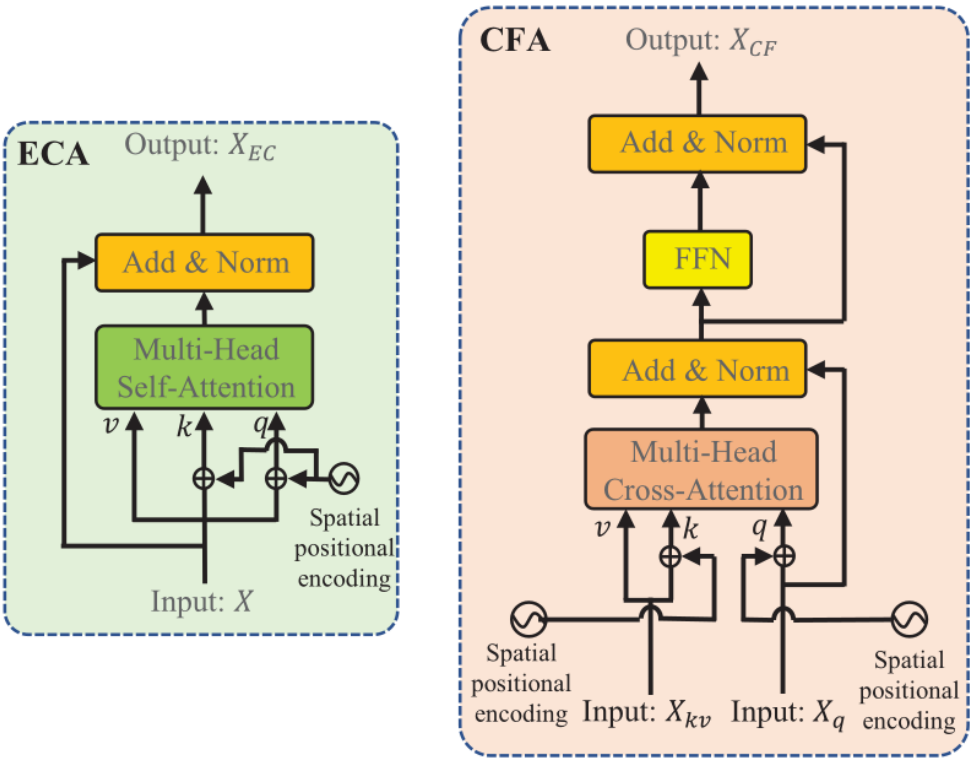

Рисунок 9 — Ego-Context Augment (ECA) and Cross-Feature Augment (CFA) modules in Transformer tracking [40].

These attention modules enable the tracker to dynamically focus on salient regions of the target, mitigating issues like distractors and occlusion.

Prediction Head: The final stage of the Transformer tracker consists of a classification and regression network that predicts the target's presence and bounding box coordinates. Compared to conventional anchor-based regression techniques used in SiamRPN [34], Transformer trackers such as TransT [40] adopt a more flexible and parameter-free prediction mechanism, improving robustness against scale variations.

Loss Function and Training Strategy

The loss function for Transformer-based tracking is typically a combination of classification and regression losses:

$$L = L_{\text{cls}} + \lambda_1 L_{\text{IoU}} + \lambda_2 L_1 \tag{15}$$

where L_{cls} is the binary cross-entropy loss for classification, L_{IoU} is the IoU loss for bounding box regression, and L_1 is the smooth l_1 loss. Training is performed using datasets such as LaSOT, GOT-10k, and TrackingNet, with an AdamW optimizer and learning rate scheduling strategies.

Performance and Advantages

Transformer-based tracking approaches have demonstrated superior performance on large-scale benchmarks such as LaSOT, TrackingNet, and GOT-10k. Experimental results from [40] indicate that TransT outperforms prior state-of-the-art trackers in terms of Average Overlap (AO) and Success Rate (SR) while maintaining real-time processing speeds (approximately 50 FPS on GPU).

Comparison with CNN-Based Tracker

Unlike CNN-based trackers, which rely heavily on handcrafted correlation operations, Transformer trackers leverage self-attention mechanisms that adaptively aggregate features across long spatial distances. This enables improved target localization, especially in scenarios involving occlusion, motion blur, and background clutter. Transformer-based methods also eliminate the need for predefined anchor boxes, leading to a more flexible and efficient tracking pipeline.

Conclusion

Transformer-based tracking represents a significant leap forward in visual object tracking, replacing handcrafted correlation operations with adaptive attention mechanisms. The ability to model long-range dependencies and adaptively fuse features makes these models particularly effective in challenging scenarios. Future work may explore hybrid architectures that integrate Transformer attention with convolutional inductive biases to further enhance tracking efficiency and robustness.

4 DATASET AND EVALUATION METRICS

4.1 UOT100 Dataset

Underwater Object Tracking (UOT100) is a benchmark dataset designed to facilitate the development and evaluation of object tracking algorithms specifically tailored for underwater environments [43; 44]. Unlike traditional tracking datasets that focus on open-air scenarios, UOT100 addresses the unique challenges posed by underwater visual data, including light attenuation, refraction, scattering, and color loss. These distortions significantly impact object visibility and tracking accuracy, making the dataset a crucial resource for advancing research in underwater object tracking.

The UOT100 dataset consists of 104 video sequences with over 74,000 manually annotated frames. These sequences encompass a diverse range of underwater conditions, including both natural and artificial environments. The dataset includes various types of distortions, such as low contrast, motion blur, occlusions, and fluctuating illumination, which are common in real-world underwater applications.

Each video sequence in the dataset is accompanied by:

– An MP4 video file capturing the object of interest.
– A ground truth annotation file containing the precise bounding box coordinates for the tracked object.

Рисунок 10 — Sample frames from the UOT100 dataset, showcasing various underwater conditions and distortions.

– A description file outlining the distortion types present in the video.
– An image sequence folder that stores individual frames for more granular analysis.

The dataset is publicly available for research purposes and serves as a standard benchmark for comparing the performance of state-of-the-art tracking algorithms in underwater conditions. It provides a foundation for assessing the robustness of correlation filter-based and deep learning-based tracking methods when applied to challenging aquatic environments

4.2 Evaluation Metrics

To objectively evaluate the performance of object tracking algorithms on the UOT100 dataset, we adopt standard evaluation metrics commonly used in the object tracking community. These metrics include **Precision**, **Success Rate (IoU-based evaluation)**, and **Frames Per Second (FPS)**, which collectively provide a comprehensive assessment of both accuracy and computational efficiency.

4.2.1 Precision

Precision measures the Euclidean distance between the predicted object center and the ground truth center across all frames in a sequence. A prediction is considered accurate if the center distance falls below a given threshold, typically set to 20 pixels as per the One-Pass Evaluation (OPE) protocol [45]. The precision plot is generated by varying the threshold from 0 to 50 pixels, showing the percentage of frames where the tracker stays within the given error range.

4.2.2 Success Rate (IoU-Based Evaluation)

Success rate is evaluated based on the **Intersection over Union (IoU)** metric, which quantifies the overlap between the predicted bounding box and the ground truth bounding box. IoU is computed as:

$$IoU = \frac{|B_{pred} \cap B_{gt}|}{|B_{pred} \cup B_{gt}|} \tag{16}$$

where B_{pred} and B_{gt} represent the predicted and ground truth bounding boxes, respectively. The success rate is derived by counting the number of frames where the IoU exceeds a predefined threshold (e.g., 0.5), and an **Area Under Curve (AUC)** score is used to rank different tracking algorithms [46].

4.2.3 Frames Per Second (FPS)

FPS measures the real-time efficiency of a tracking algorithm by calculating the number of frames processed per second. A higher FPS value indicates a more computationally efficient tracker, which is crucial for real-time underwater applications such as robotic navigation and marine surveillance [47].

By utilizing these metrics, the UOT100 benchmark provides a standardized platform for comparing different tracking approaches and identifying key challenges in underwater object tracking.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. *Qiu H.*, *Li N.*, *Li P.*, [et al.]. Boundary attention guided sparse feature learning for underwater object tracking in edge computing // ACM Transactions on Multimedia Computing, Communications and Applications. — 2024.
2. *Zhou H.*, *Kong M.*, *Yuan H.*, [et al.]. Real-time underwater object detection technology for complex underwater environments based on deep learning // Ecological Informatics. — 2024. — Vol. 82. — P. 102680.
3. *Elmezain M.*, *Saoud L.S.*, *Sultan A.*, [et al.]. Advancing Underwater Vision: A Survey of Deep Learning Models for Underwater Object Recognition and Tracking // IEEE Access. — 2025.
4. *Bhadouria A.S.* Underwater image enhancement techniques: an exhaustive study // International journal for research in applied science & engineering technology (IJRASET), ISSN. —. — P. 2321–9653.
5. *Rout D.K.*, *Subudhi B.N.*, *Veerakumar T.*, *Chaudhury S.* Walsh–Hadamard-kernel-based features in particle filter framework for underwater object tracking // IEEE Transactions on Industrial Informatics. — 2019. — Vol. 16, no. 9. — P. 5712–5722.
6. *Mathias A.*, *Dhanalakshmi S.*, *Kumar R.* Occlusion aware underwater object tracking using hybrid adaptive deep SORT-YOLOv3 approach // Multimedia Tools and Applications. — 2022. — Vol. 81, no. 30. — P. 44109–44121.
7. *Zhang C.*, *Liu L.*, *Huang G.*, [et al.]. Webuot-1m: Advancing deep underwater object tracking with a million-scale benchmark // arXiv preprint arXiv:2405.19818. — 2024.
8. *Feng W.*, *Han R.*, *Guo Q.*, *Zhu J.*, *Wang S.* Dynamic saliency-aware regularization for correlation filter-based object tracking // IEEE Transactions on Image Processing. — 2019. — Vol. 28, no. 7. — P. 3232–3245.
9. *Du S.*, *Wang S.* An overview of correlation-filter-based object tracking // IEEE Transactions on Computational Social Systems. — 2021. — Vol. 9, no. 1. — P. 18–31.

10. *Zhao S.*, *Sun K.*, *Ji Y.*, *Guo N.*, *Jia X.* Correlation filter-based object tracking algorithms // 2020 IEEE 3rd International Conference on Information Communication and Signal Processing (ICICSP). — IEEE. 2020. — P. 57–62.

11. *Lin B.*, *Zheng J.*, *Xue C.*, [et al.]. Motion-aware correlation filter-based object tracking in satellite videos // IEEE Transactions on Geoscience and Remote Sensing. — 2024. — Vol. 62. — P. 1–13.

12. *Srigowri M.* Enhancing unpaired underwater images with cycle consistent network // 2022 International Conference on Inventive Computation Technologies (ICICT). — IEEE. 2022. — P. 305–311.

13. *Lotfi F.*, *Virji K.*, *Dudek N.*, *Dudek G.* A comparison of RL-based and PID controllers for 6-DOF swimming robots: hybrid underwater object tracking // arXiv preprint arXiv:2401.16618. — 2024.

14. *Jiao L.*, *Zhang F.*, *Liu F.*, [et al.]. A Survey of Deep Learning-Based Object Detection // IEEE Access. — 2019. — Vol. 7. — P. 128837–128868.

15. *Arkin E.*, *Yadikar N.*, *Xu X.*, *Aysa A.*, *Ubul K.* A survey: object detection methods from CNN to transformer // Multimedia Tools and Applications. — 2023. — Vol. 82, no. 14. — P. 21353–21383.

16. *Yang H.*, *Xu J.*, *Lin Z.*, *He J.* LU2Net: a lightweight network for real-time underwater image enhancement // arXiv preprint arXiv:2406.14973. — 2024.

17. *Wu X.*, *Han X.*, *Zhang Z.*, [et al.]. A hybrid excitation model based lightweight siamese network for underwater vehicle object tracking missions // Journal of Marine Science and Engineering. — 2023. — Vol. 11, no. 6. — P. 1127.

18. *Pereira R.*, *Carvalho G.*, *Garrote L.*, *Nunes U.J.* Sort and deep-SORT based multi-object tracking for mobile robotics: Evaluation with new data association metrics // Applied Sciences. — 2022. — Vol. 12, no. 3. — P. 1319.

19. *Braca P.*, *Goldhahn R.*, *Ferri G.*, *LePage K.D.* Distributed information fusion in multistatic sensor networks for underwater surveillance // IEEE Sensors Journal. — 2015. — Vol. 16, no. 11. — P. 4003–4014.

20. *Tharmarasa R.*, *Kirubarajan T.*, *Hernandez M.* Large-scale optimal sensor array management for multitarget tracking // IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews). — 2007. — Vol. 37, no. 5. — P. 803–814.

21. *Uney M.*, *Stinco P.*, *Dreo R.*, [et al.]. Passive sensor fusion and tracking in underwater surveillance with the GLMB model // 2022 25th International Conference on Information Fusion (FUSION). — IEEE. 2022. — P. 1–8.

22. *Bolme D.S.*, *Beveridge J.R.*, *Draper B.A.*, *Lui Y.M.* Visual object tracking using adaptive correlation filters // 2010 IEEE computer society conference on computer vision and pattern recognition. — IEEE. 2010. — P. 2544–2550.

23. *Henriques J.F.*, *Caseiro R.*, *Martins P.*, *Batista J.* High-speed tracking with kernelized correlation filters // IEEE transactions on pattern analysis and machine intelligence. — 2014. — Vol. 37, no. 3. — P. 583–596.

24. *Henriques J.F.*, *Caseiro R.*, *Martins P.*, *Batista J.* Exploiting the circulant structure of tracking-by-detection with kernels // Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part IV 12. — Springer. 2012. — P. 702–715.

25. *Danelljan M.*, *Häger G.*, *Khan F.*, *Felsberg M.* Accurate scale estimation for robust visual tracking // British machine vision conference, Nottingham, September 1-5, 2014. — Bmva Press. 2014.

26. *Danelljan M.*, *Häger G.*, *Khan F.S.*, *Felsberg M.* Discriminative scale space tracking // IEEE transactions on pattern analysis and machine intelligence. — 2016. — Vol. 39, no. 8. — P. 1561–1575.

27. *Felzenszwalb P.F.*, *Girshick R.B.*, *McAllester D.* Cascade object detection with deformable part models // 2010 IEEE Computer society conference on computer vision and pattern recognition. — Ieee. 2010. — P. 2241–2248.

28. *Danelljan M.*, *Shahbaz Khan F.*, *Felsberg M.*, *Van de Weijer J.* Adaptive color attributes for real-time visual tracking // Proceedings of the IEEE conference on computer vision and pattern recognition. — 2014. — P. 1090–1097.

29. *Wang N.*, *Li S.*, *Gupta A.*, *Yeung D.-Y.* Transferring rich feature hierarchies for robust visual tracking // arXiv preprint arXiv:1501.04587. — 2015.

30. *Danelljan M.*, *Hager G.*, *Shahbaz Khan F.*, *Felsberg M.* Learning spatially regularized correlation filters for visual tracking // Proceedings of the IEEE international conference on computer vision. — 2015. — P. 4310–4318.

31. *Held D.*, *Thrun S.*, *Savarese S.* Learning to track at 100 fps with deep regression networks // Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14. — Springer. 2016. — P. 749–765.

32. *Jia Y.*, *Shelhamer E.*, *Donahue J.*, [et al.]. Caffe: Convolutional architecture for fast feature embedding // Proceedings of the 22nd ACM international conference on Multimedia. — 2014. — P. 675–678.

33. *Krizhevsky A.*, *Sutskever I.*, *Hinton G.E.* AlexNet - ImageNet Classification with Deep Convolutional Neural Networks. —.

34. *Li B.*, *Yan J.*, *Wu W.*, *Zhu Z.*, *Hu X.* High performance visual tracking with siamese region proposal network // Proceedings of the IEEE conference on computer vision and pattern recognition. — 2018. — P. 8971–8980.

35. *Bertinetto L.*, *Valmadre J.*, *Henriques J.F.*, *Vedaldi A.*, *Torr P.H.* Fully-convolutional siamese networks for object tracking // Computer vision–ECCV 2016 workshops: Amsterdam, the Netherlands, October 8-10 and 15-16, 2016, proceedings, part II 14. — Springer. 2016. — P. 850–865.

36. *Li B.*, *Wu W.*, *Wang Q.*, [et al.]. Siamrpn++: Evolution of siamese visual tracking with very deep networks // Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. — 2019. — P. 4282–4291.

37. *Guo D.*, *Wang J.*, *Cui Y.*, *Wang Z.*, *Chen S.* SiamCAR: Siamese fully convolutional classification and regression for visual tracking // Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. — 2020. — P. 6269–6277.

38. *Yan B.*, *Peng H.*, *Wu K.*, [et al.]. Lighttrack: Finding lightweight neural networks for object tracking via one-shot architecture search // Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. — 2021. — P. 15180–15189.

39. *Lu J.*, *Li S.*, *Guo W.*, [et al.]. Siamese graph attention networks for robust visual object tracking // Computer Vision and Image Understanding. — 2023. — Vol. 229. — P. 103634.

40. *Chen X.*, *Yan B.*, *Zhu J.*, [et al.]. Transformer tracking // Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. — 2021. — P. 8126–8135.

41. *Vaswani A.*, *Shazeer N.*, *Parmar N.*, [et al.]. Attention is all you need // Advances in neural information processing systems. — 2017. — Vol. 30.

42. *He K.*, *Zhang X.*, *Ren S.*, *Sun J.* Deep residual learning for image recognition // Proceedings of the IEEE conference on computer vision and pattern recognition. — 2016. — P. 770–778.

43. *Kezebou L.*, *Oludare V.*, *Panetta K.*, *Agaian S.S.* Underwater object tracking benchmark and dataset // 2019 IEEE International Symposium on Technologies for Homeland Security (HST). — IEEE. 2019. — P. 1–6.

44. *Panetta K.*, *Kezebou L.*, *Oludare V.*, *Agaian S.* Comprehensive underwater object tracking benchmark dataset and underwater image enhancement with GAN // IEEE Journal of Oceanic Engineering. — 2021. — Vol. 47, no. 1. — P. 59–75.

45. *Wu Y.*, *Lim J.*, *Yang M.-H.* Online object tracking: A benchmark // Proceedings of the IEEE conference on computer vision and pattern recognition. — 2013. — P. 2411–2418.

46. *Kristan M.*, *Leonardis A.*, *Matas J.*, [et al.]. The sixth visual object tracking vot2018 challenge results // Proceedings of the European conference on computer vision (ECCV) workshops. — 2018. — P. 0–0.

47. *Nam H.*, *Han B.* Learning multi-domain convolutional neural networks for visual tracking // Proceedings of the IEEE conference on computer vision and pattern recognition. — 2016. — P. 4293–4302.