Energy-Efficient Image Recognition System for Marine Life

H. Seckin Demir, Jennifer Blain Christen, *Member, IEEE,* and Sule Ozev, *Member, IEEE,*

***Abstract*—This paper focuses on designing an energy-efficient image recognition system for marine monitoring. One of the main challenges of an underwater imaging system is the strict power consumption constraints due to the limited on-site resources. Considering the need for continuous operation in different water turbidity levels and background illumination conditions, an energy-efficient approach is needed for effective utilization of the resources. In this work, we propose a recognition framework that will adaptively adjust the system parameters, such as camera frame rate and LED illumination level, based on the environment conditions to optimize the energy consumption while ensuring a high recognition accuracy. The first part of the proposed decision system contains the convolutional neural network (CNN) based animal recognition block which is used for obtaining the confidence level for a single frame. The second part is the adaptive decision block that dynamically changes the system parameters and combines the results of the recognition block for multiple frames based on the environment conditions. In our experiments, we have used nearly 8000 underwater images for training and testing the single frame recognition block and used nearly 200 different video sequences for training and testing the adaptive decision block. Based on measurements of a hardware framework composed of a Raspberry Pi 3 Model B, a Pi NoIR Camera v2.1, and 850nm LEDs, the proposed system achieves up to 92.7% energy savings with a comparable recognition performance by dynamically changing the frame rate and emitted light intensity based on water turbidity and background illumination level.**

***Index Terms*—underwater object recognition, convolutional neural networks, energy-efficiency**

I. Introduction

Artisanal gillnet fisheries have been an important source for food sustaining coastal populations [1]. However, unintended capture of different animal species such as sea turtles, sharks and other marine mammals can result in decline in the population of these species and damage the ecosystem [2]–[4]. This unintended capture, termed bycatch, presents a threat especially to green turtles, since recent studies estimate the number of sea turtles killed in fisheries as hundreds of thousands per year [5]–[8]. Furthermore, these incidents cause significant financial losses to coastal communities by damaging the fishing gears or mandating location changes from profitable regions [9].

To reduce the bycatch of sea turtles, various approaches have been developed over the years [10]–[12]. These studies aim to develop novel fishing gear designs with visual or acoustic stimuli to prevent bycatch. However, the effectiveness of the designs should be maximized by experimenting with different stimuli and analyzing the associated response given by sea turtles. Performing this analysis manually by a human operator is not practical, since it will require monitoring the fishing areas for long periods and observing the sea turtle behaviours by altering the stimuli at the right time. Luckily, with the developments in vision-based object recognition techniques, an underwater animal recognition system can be designed to help collect the sorely needed data.

Object recognition is a widely studied problem for many applications in various fields [13]–[17]. Developments in machine learning have contributed to achieving very successful recognition especially in large datasets, making the deep neural networks a de facto standard for object recognition applications [18]–[21]. These developments helped the underwater animal recognition task as well [14], [22]–[25]. In these studies, the main focus is typically improving the recognition accuracy of the system. However, one of the most significant challenges for an underwater recognition system is the energy limitation when real-time operation with an underwater hardware setup is required. For such a system, camera parameters and emitted light intensity should be carefully controlled to use power resources efficiently. However, these parameters have a direct impact on the recognition performance. Therefore, a high-level decision mechanism that will consider both the recognition performance and energy efficiency is required. To this end, we propose an energy efficient underwater recognition framework that will optimize the system parameters based on the changing environment conditions to achieve lower power consumption while obtaining a successful recognition rate.

The proposed framework contains a deep artificial neural network for obtaining the confidence level for sea turtle recognition task. The output of this network for multiple frames is processed along with the environmental conditions, such as water turbidity and background illumination level, to achieve a final decision for the recognition of sea turtles. Based on the environmental conditions, the decision block also dynamically changes the system parameters to decrease power consumption and achieve the best possible recognition rate under these conditions.

The organization of this paper is as follows: Section II summarizes the prior work in artificial neural networks and other machine learning approaches for the object recognition task. This section also includes various approaches for the recognition of underwater animal species. In Section III, we give an overview of the proposed optimization framework and introduce the functional sub-blocks. Section IV explains our

This article was presented in the International Conference on Hardware/Software Codesign and System Synthesis 2020 and appears as part of the ESWEEK-TCAD special issue.

H.S. Demir, J. Blain Christen and S. Ozev are with the Department of Electrical, Computer, and Energy Engineering, Arizona State University, Tempe, AZ, 85287, USA (e-mail: {hdemir, jennifer.blainchristen, sule.ozev}@asu.edu)

This work is supported by the National Science Foundation with Award Number CPS-1837473.

methodology for training the deep artificial neural network and optimizing the system parameters for the proposed adaptive decision mechanism. Section V reports the experimental results, while Section VI provides the concluding remarks.

II. Prior Work

Researchers have long used autonomous underwater camera systems to observe and understand marine life [26], [27]. In such underwater image capture systems, the camera is typically placed in a fixed location and works at a fixed frame rate and background illumination [26], [27]. The results from the captured images are analyzed and adjustments are made off-line [26], [27]. Various image processing and machine learning methods have been proposed for underwater animal recognition over the past few decades. Early approaches generally require a controlled environment for performing the task. For example, in [28], a recognition method based on the shape and the color of the sample is proposed for fish species under laboratory conditions. However, recognition of underwater animal species in the natural environment is a more challenging problem, since the images captured in these conditions will contain illumination variations, water turbidity and background clutter. The recognition task becomes even more challenging as there is no constraint in the orientation of the animals of interest. Early methods using texture patterns and shapes for the recognition in natural environment achieve good results only for the highly distinguishable shapes and patterns [29], [30].

With the developments in machine learning, various supervised and unsupervised methods have been employed in underwater animal recognition problem. Some of the early approaches utilized principal component analysis (PCA) [13] or linear discriminant analysis (LCA) [31] for the task and obtained moderate recognition rates. Sparse representation classification [14], Gaussian mixture models (GMM), and support vector machines (SVM) [32], [33] are among the other machine learning-based approaches proposed later for the recognition of underwater animal species. Although these methods achieved more and more successful results over time, remarkable leaps in the object recognition rates were obtained after the utilization of deep artificial neural networks. In [34], Krizhevsky et al. achieved nearly a 10% reduction in the error rate for the general object recognition task by using the Convolutional Neural Network (CNN) architecture on the ImageNet [21] dataset. This successful result quickly turned CNNs into a commonly used architecture for a wide range of object recognition applications.

Convolution has already been a commonly used operation for different computer vision applications such as denoising or edge detection. In [17], LeCun demonstrated that the filter coefficients in the convolution operations can be automatically learned from data using neural networks. Such an architecture typically contains convolution layers, nonlinear activation layers, pooling layers and fully connected output layers. Convolution layers in the architecture are used for extracting distinguishing features to recognize various object classes. While the first layers obtain the low level features such as edges and corners, deeper layers extract more complex and higher level features. With the increasing number of layers, a proper architecture will have higher representation power and superior recognition capability. However, the training process becomes harder for deeper networks due to the unstable or vanishing gradient problem [35]. In [36], He et al. proposed the concept of Residual Learning using Identity mappings through by-pass connections. This approach provided an alternative route for the gradient to flow in the back-propagation stage, making it possible to build deeper networks with fewer trainable parameters than traditional CNNs. The ResNet architecture achieved superior recognition performance on ImageNet [21] dataset when compared to the older approaches. Furthermore, the number of weights in the network architecture is lower for ResNet ($\sim$25.5M for ResNet50) compared to other commonly used network architectures, such as AlexNet ($\sim$61M) [34] and VGG-16 ($\sim$138M) [37]. Therefore, we have utilized this network design for our sea turtle recognition task.

Another issue with increasing the depth of a neural network is the overfitting problem. Since the network becomes more complex with higher number of layers and coefficients, it can fit well to training samples, but perform poorly in the unseen dataset for a limited training dataset. To overcome this problem, transfer learning is a broadly used approach [38], where network coefficients trained for a certain application are partially or fully utilized for another task. In our design, we used a model with coefficients pre-trained on the ImageNet [21] dataset.

III. Overview of the Recognition System

The proposed recognition system has several components as shown in Figure 1. The fundamental functional blocks are explained in the following subsections.

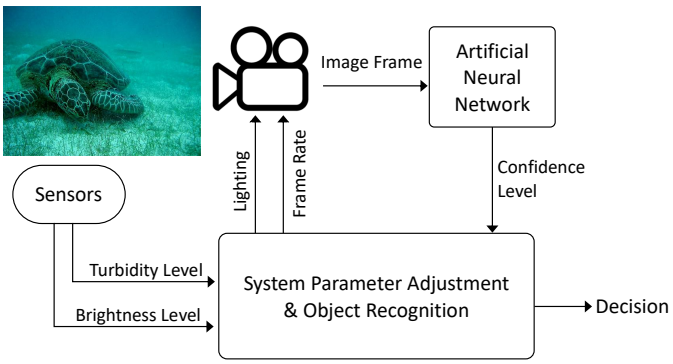


Fig. 1: Block diagram of the recognition system

A. Artificial Neural Network

A significant part of the recognition system is the deep neural network architecture that processes the image frames and computes a confidence level for the animal species of interest. Convolutional neural networks have been commonly used for the object recognition task for different applications [18]–[20], [39]. We have utilized a model that is pre-trained

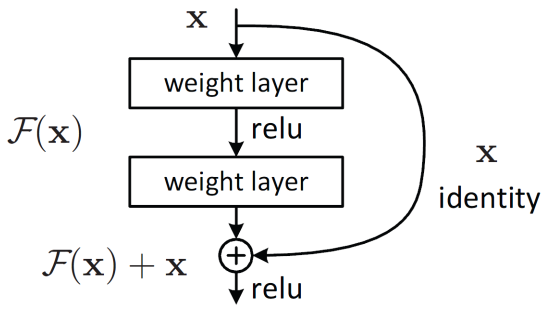


Fig. 2: A building block of residual network [36]

on the ImageNet [21] dataset. Normally, this dataset is not created for underwater animal species, but there are nearly 1.2 million images consisting of a wide range of objects. Training deep artificial neural networks on such a large dataset that includes diverse number of sample foreground and background objects makes it possible to extract representative features for the recognition of various objects. Therefore, a proper network architecture can learn color, texture, or shape patterns and achieve generalization power, if there is sufficient number of samples and variety in the training set. Due to this generalization power, such networks can be utilized in the recognition task for other new datasets. Even if these datasets are not used in training the neural network, learnt features are representative enough to be utilized for the recognition of unseen data samples. Some of the well known networks that are used for feature extraction include AlexNet [34], VGGNet [37] and Resnet [36]. In our implementation, we have utilized the Deep Residual Network (ResNet-50) architecture [36] and employed the transfer learning approach [40] for calculating the network coefficients.

A building block of residual networks is shown in Figure 2. The weight layers include convolution kernels as found in a typical CNN architecture. In addition to these convolutional layers, the main advantage of this structure is the bypass connections between the inputs and outputs of the layers. Normally, outputs of the convolution layers in CNNs are used as inputs for the next layers. However, in this structure, bypass links propagate the effect of extracted features in the first layers to the next layers through direct connections. This architecture not only helps combining a larger set of features in the deeper layers, it also allows for increasing the depth of the network for higher recognition accuracy. Another issue with deep CNNs is unstable or vanishing gradient problem in the backpropagation process [35]. These bypass connections also help prevent the vanishing gradient problem, since the gradient is transmitted directly over the bypass connection in addition to the gradient through the branch.

In our design, we have replaced the last 2 fully-connected layers of the utilized ResNet-50 architecture pre-trained on ImageNet [21] dataset. The coefficients of these 2 fully connected layers are trained using our dataset for the sea turtle recognition task.

B. Adaptive Decision Block

The trained network can recognize images very accurately under ideal conditions, such as clear water and significant background illumination. However, underwater conditions can change dramatically, reducing the fidelity of the image. Background illumination can change depending on the time of day, depth, and water conditions. In order to obtain clear images, the camera system includes LED lighting that can provide additional background illumination. However, LEDs consume significant power and their use when the background conditions are ideal can lead to unnecessary draining of the precious resources. Similarly, water turbidity can change dynamically, inducing a fog-like effect on the images. Luckily, we can use multiple images to increase the likelihood of making the correct decision by capturing sequential images. This solution also leads to increased use of limited resources. Thus, it is necessary to adjust the background illumination as well as the number of images processed and capture rate of the image frames.

The proposed adaptive decision block provides a decision by combining recognition results for multiple frames. Based on the turbidity and background illumination level, the decision block adjusts the system parameters, namely camera frame rate and the intensity of the emitted light. This block also changes the internal decision parameters for achieving a high accuracy for the recognition task. The purpose of the system is to obtain a sufficient recognition accuracy while keeping the energy consumption at a minimum level due to the power constraints.

IV. Optimization Methodology

In our approach, we have focused on developing an energy efficient marine animal detection system. Since the main energy consuming components of the detection system are the camera and LED lighting, we aim to minimize these energy costs based on the environment conditions. In order to optimize the parameters based on the changing conditions, we have emulated the effect of system parameter settings (e.g. LED intensity) as well as environmental effects (background illumination, water turbidity) on the resulting images by modifying the image settings in the existing set. We have also measured the power consumption patterns of changing LED intensity and camera frame rate for the experimental hardware and generated a model to be used in the optimization process.

A. Dataset

We have used two different datasets for training the deep convolutional network and optimizing the parameters of the decision block.

1) Single Frame Recognition: The first dataset contains more than 8000 underwater images to train the network for the task of recognizing sea turtle with a single frame. Nearly half of the images are positive samples which include sea turtles from different angles and with different backgrounds. The other half of the dataset are negative samples which contains different underwater scenes with or without different animal

species. A few examples from the single frame recognition dataset are given in Figure 3.

We augmented the data by using the cropped and rotated versions of the images. The number of samples in the dataset is also increased by adding different levels of a synthetically generated turbidity effect to the images. We employed a similar model explained in [41] to extend our dataset using the properties of turbidity. In this model, effects of color distortion, blur and contrast degradation are added through the following equations.

$$I_G(x,y) = (1-\alpha_B)I(x,y) + (\alpha_B)g(x,y) \tag{1}$$

where I_G is the blurred image and α_B is the parameter that controls the blurring amount by changing the weights of the original image I and the Gaussian filter result g. Eqn. 2 shows the computation of g.

$$g(x,y) = \sum_{k=-N/2}^{N/2} \sum_{l=-N/2}^{N/2} I(x-k, y-l)h(k,l) \tag{2}$$

where the coefficients of the filter, h, are calculated using the Eqn. 3.

$$h(x,y) = \begin{cases} K\exp(-\frac{x^2+y^2}{2\sigma^2}) & -N/2 \leq x,y \leq N/2 \\ 0 & elsewhere \end{cases} \tag{3}$$

where K is normalization factor and σ^2 is the variance of the Gaussian filter. In addition to blur, we also emulate the color distortions and contrast degradation in the turbidity model [41] as given in Eqn. 4.

$$I_T(x,y) = (1-\alpha_C)I_G(x,y) + \alpha_C C_T \tag{4}$$

(a) Positive samples

(b) Negative samples

Fig. 3: Sample images from the dataset

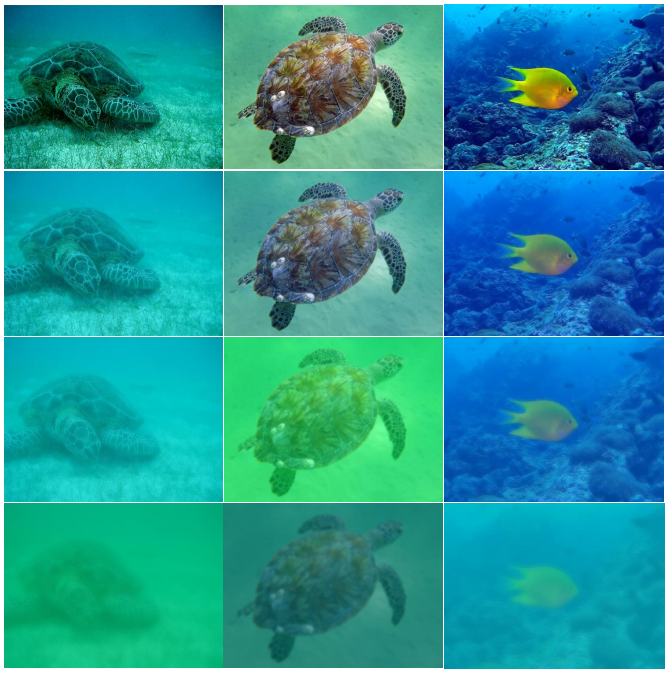


Fig. 4: Sample images for synthetically generated turbidity effect. Top row shows the original images while the second, third, and fourth rows correspond to turbidity level 1, 2, and 3 respectively

where α_C is the parameter for controlling the contrast degradation and color distortion level. C_T represents the RGB (red, green, blue) values for turbidity effect. Using this model, we have emulated three different levels of turbidity with different parameter sets and generated the turbid image dataset. In order to build a variety of images and to not overfit a single effect, we perturbed the parameters for different images. For example, for the C_T, red value is 0, while green and blue values are uniformly sampled in the range $(0.3, 1.0)$. This gives a random color with green and blue components for the turbidity effect. We also used the value ranges given in Table I for the other parameters.

TABLE I: Parameter values for different turbidity levels.

Turbidity Level	α_B	α_C	σ
1	0.32 ± 0.07	0.3 ± 0.1	4 ± 1
2	0.57 ± 0.07	0.5 ± 0.1	7 ± 1
3	0.82 ± 0.07	0.7 ± 0.1	10 ± 1

Examples from the synthetically generated turbidity images are given in Figure 4. The enhanced single frame recognition dataset is utilized for training the residual deep neural network to achieve a high accuracy in the turtle recognition task. 70% of the images in this dataset are used for training the network, while 20% are used for validation and 10% are used for testing results. The samples are selected randomly and there is no overlap between the training, validation, and test samples to eliminate any bias in performance evaluation.

2) Scenario-based Recognition: The proposed framework aims to achieve high recognition accuracy while minimizing energy consumption by adapting to environmental conditions.

Fig. 5: Different brightness levels are simulated using the reference images taken in the lab conditions

In order to enable this optimization, the decision block should dynamically adjust the system parameters and provide a recognition output by leveraging multiple frames. Therefore, the parameters utilized in the system should be trained and tested for different environment conditions with various turbidity levels, background illumination levels, and scene content. For this purpose, we prepared more than 200 video sequences, each of which includes nearly 300 frames (corresponding to 10 seconds of video with 30 fps). For different environmental conditions, we also synthetically generated variations of the image sequences with different background illumination and turbidity levels.

The turbidity effect is emulated using the same model and parameters explained in the previous section. In order to emulate the effect of background illumination and LED lighting, we used a controlled lab setup consisting of a camera, LEDs, and various objects to capture at a certain distance. Under dim background illumination, we captured images of the same scene with different LED lighting levels without changing the camera settings (Shutter speed, aperture and ISO level). Using the intensity values of these images as reference, we applied a transformation to the original dataset to generate images with these intensity levels. Some of the example images generated using this approach are given in Figure 5. Our goal in this process is to mimic varying background illumination level and different LED intensity levels by adjusting parameters of the existing underwater images.

B. Multi-frame Decision Algorithm

As explained above, the Deep Residual Network in the system gives a confidence level for the recognition of the object of interest in a single frame. The adaptive decision

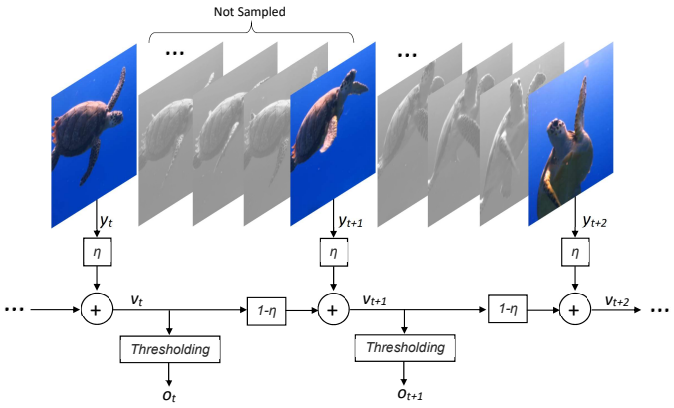


Fig. 6: Overview of the decision mechanism

block, on the other hand, uses multiple frame results to provide the final recognition decision. A brief overview of the system is depicted in Figure 6. As it can be seen from the figure, some of the frames are not sampled and processed based on the dynamically adjusted frame rate. For the processed frames, confidence levels obtained from the neural network is combined in the temporal domain. Let us assume that we have the confidence level y_t for the time instance t. This value is averaged with the previous confidence levels to obtain a filtered and more accurate result v_t as shown in Eqn. 5. Here, v_t is the result of filtered confidence level. v_t is compared with a detection threshold, T_D, to achieve a binary decision for a given image set as given in Eqn. 6.

$$v_t = (1-\eta)v_{t-1} + \eta y_t \tag{5}$$

$$o_t = \begin{cases} 1 & v_t \geq T_D \\ 0 & v_t < T_D \end{cases} \tag{6}$$

In order to reach a final decision, a certain number of consecutive detections (T_F) are expected. These steps are summarized in Algorithm 1.

C. Energy Optimization

In the previous subsection, we explained our algorithm for maximizing the number of correct decisions for the object recognition task using multiple frames. However, we also need to consider the energy consumption, while keeping a successful recognition rate. In the proposed system, camera frame rate and LED illumination level are two main sources of the energy consumption that we can control. Therefore, instead of minimizing only the error rate, we define a cost function that includes energy costs of illumination level and camera frame rate as well. The defined cost function is given in Eqn. 7.

$$J(\theta) = (1 - Acc(\theta))^2 + \lambda(c_f(\theta)^2 + c_l(\theta)^2) \tag{7}$$

where the parameter set, θ, includes camera frame rate, f, and LED illumination level, l, as well as η, T_D, and T_F

Algorithm 1 Multi-frame Decision Algorithm

Input: Confidence Level: y_t; Threshold Values: T_D, T_F; Filter coefficient: η; Number of frames: N

Initialize detection counter $C_D \leftarrow 0$
Initialize filtered confidence level $v_0 \leftarrow y_0$
for $t = 1 \rightarrow N$ **do**
    Obtain confidence level y_t
    $v_t = \eta y_t + (1-\eta) v_{t-1}$
    if $v_t \geq T_D$ **then**
        $o_t \leftarrow 1$
        $C_D \leftarrow C_D + 1$
        if $C_D \geq T_F$ **then**
            Declare Recognition $r_t \leftarrow 1$
        end if
    else
        No Recognition $r_t \leftarrow 0$
        $o_t \leftarrow 1$
        $C_D \leftarrow 0$
    end if
end for

Return: Recognition Result: r_t

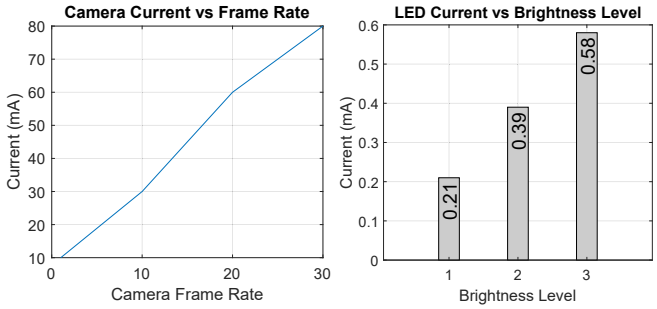
introduced in the multi-frame decision algorithm. Given this parameter set, $(1-Acc)$ corresponds to error rate while c_f and c_l represent the cost of camera frame rate and LED lighting respectively. The coefficient λ is used for adjusting the trade-off between error rate and energy costs. In this equation, c_f and c_l should be chosen based on the energy consumption levels of the camera frame rate and LED illumination.

While our optimization method is not limited to specific hardware, we adjusted these parameters on an example hardware setup without loss of generality. In order to observe the energy costs of these parameters, we have used Raspberry Pi 3 Model B [42], Pi Camera v2.1, 850nm LEDs and captured images with different frame rates and illumination levels. Figure 7 shows the measured current drawn by the camera and LEDs under different settings.

Note that brightness levels in Figure 7b are the levels we used for creating our dataset shown in Figure 5. Thus, we achieve a correspondence between the utilized cost function and synthetically generated brightness effect in our dataset. Since the energy costs will be proportional to the current values drawn by the camera and LEDs, we set c_f and c_l to the associated current values for given f and l in the parameter set.

In the optimization process, λ should be chosen such that a good balance between error rate and energy costs is achieved. Setting this value too large might cause an insufficient recognition rate while setting it too low could lead to high energy consumption. In our experiments, we set this value to 3×10^{-7} based on our power measurements and desired recognition rates.

Now that we have the cost function, we need to find the parameters that will minimize J under different conditions. Since we have many parameters with infinitely many possible values, it is impossible to calculate the cost function at every point in the solution space. Moreover, the analytical solution is not possible, since a part of the cost function comes from the performance of a deep neural network consisting of millions of parameters. Furthermore, the cost function possibly includes multiple local minima as well. Since the problem is a discrete optimization problem with a large configuration space and local minima, we have utilized Simulated Annealing [43] for hill climbing purposes.



(a) (b)

Fig. 7: Current drawn by the camera and LEDs. a) Camera frame rate vs current b) LED illumination level vs current (Reference voltage used for both hardware is 5V)

V. Experimental Results

In our experiments, we first trained the last 2 fully-connected layers of the pre-trained ResNet-50 network using our single frame dataset. After that, we utilized this network as a part of our multi-frame recognition system and trained the parameters of the decision block using the scenario-based recognition dataset. In our experiments, we have used accuracy and F_1-score as the performance measures. The definitions of these measures are given in Eqn. 8 and Eqn. 9.

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \tag{8}$$

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall} \tag{9}$$
$$precision = \frac{TP}{TP + FP}$$
$$recall = \frac{TP}{TP + FN}$$

where TP, TN, FP, and FN represent the number of true positives, true negatives, false positives, and false negatives respectively.

A. Single Frame Recognition Results

We conducted a set of experiments to observe the performance change when we include the synthetically generated turbid water images in the training dataset. In these experiments, we first trained the network using the original dataset and obtained the accuracy results for the test images with different turbidity levels. Then, we repeated the experiments

by including turbid water images in the training dataset and obtained the recognition accuracy for the same test set. The results are shown in Table II.

TABLE II: Recognition accuracy obtained using different training sets

	Test Set			
Training Set	Original	T. L1	T. L2	T. L3
Original	94.9	92.9	91.8	72.8
Turbid L3	94.6	92.7	89.5	77.0
Original + Turbid L3	97.2	96.0	94.1	82.0

As the Table II shows, we achieve the best results by including both the original dataset and synthetically generated turbidity dataset in the training step. Accuracy and F_1-score of the Deep Residual Neural Network for the single frame recognition dataset are given in the Table III.

TABLE III: Performance of the Deep Residual Neural Network on the single image dataset

Dataset	Acc	F_1-measure
Original	97.2	96.9
Turbid L1	96.0	96.1
Turbid L2	94.1	93.8
Turbid L3	82.0	81.7

Table III shows that the recognition performance degrades with the increasing water turbidity level. This is an expected result, since the synthetically generated turbidity effect decreases the contrast, distorts the color, and blurs the details, causing the degradation in visual features used for the recognition task.

B. Scenario-Based Recognition Results

In these experiments, we make a decision for a scenario which contains multiple frames rather than a single frame. Therefore, we should first clarify the definitions of true positive, true negative, false positive, and false negative for the scenario-based experiments. In these runs, we simulate a case where we want to obtain a positive decision, i.e., a sea turtle enters the observed area at some point. Therefore, a positive sample in this dataset is an image sequence where there exist one or more frames containing the animal of interest. That is, a positive sample in this dataset may not necessarily contain the sea turtle for all the frames; yet the expected decision for this sequence is still positive. A negative sample, on the other hand, does not contain the object of interest in any of the frames throughout the sequence, meaning the expected outcome from the decision block is negative. A positive decision is given when there exist a moment, t, in the image sequence where the recognition result, r_t, is 1. For a negative decision, on the other hand, r_t must be 0 for all the frames throughout the image sequence. Hence, the definitions of TP, TN, FP, and FN are made based on this decision per each video sequence.

In Eqn. 7, we defined the cost function based on the system and environmental parameters. We have used the power consumption measurements in the cost function and altered the λ parameter to find a good balance between the recognition accuracy and power cost. We aimed to have a worst case recognition accuracy around 90% and obtained it for the λ value of 3×10^{-7}. To minimize this cost function, we have searched our parameter space using the Simulated Annealing method and used the obtained Acc, c_f and c_l values in the equation. Experiments on the synthetically generated turbidity and illumination datasets gave us the parameter values in Table IV for the optimized system.

TABLE IV: Parameter optimization result for the scenario-based recognition dataset

	Water Turbidity Level			
	0	1	2	3
Detection Threshold (T_D)	0.83	0.79	0.69	0.15
Consecutive Rec. Thr (T_F)	1	1	1	14
Output Filtering Weight (η)	0.73	0.71	0.63	0.45
Camera Frame Rate (fps) (f)	2.73	1.11	1.15	4.29
LED Illumination Level (l)	2	3	3	4
Accuracy (Acc)	99.03	99.51	96.6	89.81

When the optimization result for higher water turbidity levels is examined, it can be observed that the decision block combines the results of more frames by increasing the camera frame rate, f, decreasing the filter weight, η, and increasing the number of consecutive recognition threshold, T_F. It is also observed that the detection threshold, T_D, is decreased and LED illumination level is increased as the water turbidity level increases. These results show that the decision block adaptively changes the system parameters such that the parameters associated with higher power consumption, namely f and l are minimized when the environment conditions are better in terms of image quality. In other words, the recognition task is performed with lower frame rate and LED levels, which results in lower energy consumption. It might be thought that decreasing the sampling rate could affect the camera's ability to capture the target object. However, for the bycatch reduction problem specifically, it is unlikely for the target animal to be traveling at high speeds (e.g. $>$10m/s). This is also observed in the video datasets that we processed. The results also demonstrate that the adaptive parameter selection scheme enables the decision system to leverage multiple frame results and achieve more accurate results under challenging environment conditions.

In order to evaluate the energy savings of our optimization approach, we first measured hardware parameters of a sample image capture system, namely, Raspberry Model 3 B [42] with Pi Camera v2.1. For this system, we have run various experiments and determined that the major energy consuming tasks are image capture and processing and background LED illumination. We have determined that the energy consumption of these tasks is significantly higher than baseline energy consumption of the Raspberry Pi Model 3 B. In these hardware measurements, we use discrete levels of background LED illumination, ranging from 1 LED unit to 3 LED units. As expected, the energy consumption of the background illumination subsystem is linearly related to the illumination level. Furthermore, we have measured the current consumption of the hardware setup with Pi Camera v2.1 under various frame rates to determine the dependency of energy consumption with respect to the number of captured and processed frames. In this case, we observe a significant correlation which can be

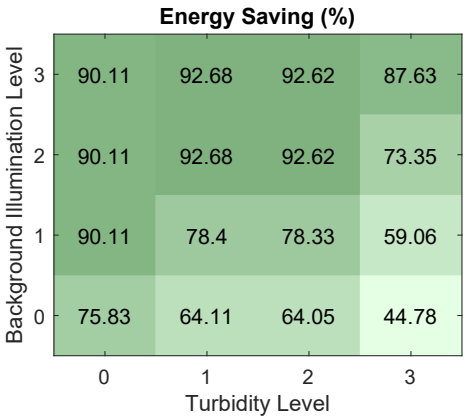


Fig. 8: Energy saving percentage of the proposed method compared to base frame rate and LED lighting level under different environment conditions

modeled as a polynomial function. The picture of the hardware measurement results along with the summary of the current consumption patterns are given in Figure 7.

In order to evaluate the energy consumption benefits of the proposed optimization system, we define a baseline recognition system that works at 30fps, which is the default rate of the camera and at maximum background LED illumination level, which is the default illumination level of the camera background light. We repeated our experiments with this baseline system and compared our accuracy and energy consumption levels. Using the measurements given in Figure 7 and the results given in Table IV, we calculated the energy saving percentage of the proposed approach for different environmental conditions compared to the baseline recognition system. Figure 8 shows that the proposed approach provides up to 92.7% energy consumption savings by the camera and LEDs depending on the conditions. As we can expect, the optimizer utilizes more LED units and higher frame rates for more challenging conditions with higher turbidity levels and lower background illumination. Even in the worst case scenario, the optimized system achieves 44.8% energy saving compared to baseline recognition system. We also compared the performance of proposed optimized recognition system and the baseline recognition system on the scenario-based recognition dataset. Figure 9 shows that the proposed energy-efficient decision scheme provides comparable accuracy and F_1-score with the baseline recognition system. Another important observation is that the recognition rates of the scenario based experiments for all turbidity levels are higher than the corresponding single frame recognition rates due to the adaptive decision block that successfully combines multiple frame results.

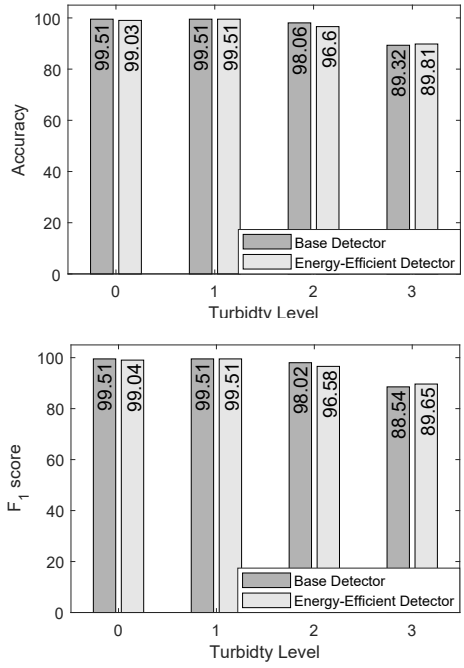


Fig. 9: Accuracy and F_1 score comparison of the base detector and the proposed energy-efficient detection system

VI. Conclusion

Automated recognition of underwater animals is an important task that can help marine life preservation. However, one of the main challenges of an underwater imaging system is the limited on-site power resources, including the overall battery charge, which requires an energy-efficient approach. In this work, we propose an energy-efficient underwater animal recognition framework that adaptively optimizes the system parameters based on the environmental conditions to decrease power consumption while still providing a successful recognition result. Our experiments on the scenario-based dataset with synthetically generated turbidity and illumination effects show that the proposed approach provides up to 92.7% energy savings by changing the system and decision parameters based on the environmental conditions. Results also show that the proposed energy-efficient approach yields comparable recognition performance with the baseline recognition system that runs with full frame rate and high emitted light intensity.

REFERENCES

[1] R. Willman, K. Kelleher, R. Arnason, and N. Franz, "The sunken billions: the economic justification for fisheries reform," no. no, 2009.

[2] S. H. Peckham, D. M. Díaz, A. Walli, G. Ruiz, L. B. Crowder, and W. J. Nichols, "Small-scale fisheries bycatch jeopardizes endangered pacific loggerhead turtles," *PloS one*, vol. 2, no. 10, 2007.

[3] S. H. Peckham, D. Maldonado-Diaz, V. Koch, A. Mancini, A. Gaos, M. T. Tinker, and W. J. Nichols, "High mortality of loggerhead turtles due to bycatch, human consumption and strandings at baja california sur, mexico, 2003 to 2007," *Endangered Species Research*, vol. 5, no. 2-3, pp. 171–183, 2008.

[4] J. A. Estes, J. Terborgh, J. S. Brashares, M. E. Power, J. Berger, W. J. Bond, S. R. Carpenter, T. E. Essington, R. D. Holt, J. B. Jackson *et al.*, "Trophic downgrading of planet earth," *science*, vol. 333, no. 6040, pp. 301–306, 2011.

[5] B. P. Wallace, R. L. Lewison, S. L. McDonald, R. K. McDonald, C. Y. Kot, S. Kelez, R. K. Bjorkland, E. M. Finkbeiner, S. Helmbrecht, and L. B. Crowder, "Global patterns of marine turtle bycatch," *Conservation letters*, vol. 3, no. 3, pp. 131–142, 2010.

[6] F. Humber, B. J. Godley, and A. C. Broderick, "So excellent a fishe: a global overview of legal marine turtle fisheries," *Diversity and Distributions*, vol. 20, no. 5, pp. 579–590, 2014.

[7] P. Casale, "Sea turtle by-catch in the mediterranean," *Fish and Fisheries*, vol. 12, no. 3, pp. 299–316, 2011.

[8] J. Alfaro-Shigueto, J. C. Mangel, F. Bernedo, P. H. Dutton, J. A. Seminoff, and B. J. Godley, "Small-scale fisheries of peru: a major sink for marine turtles in the pacific," *Journal of Applied Ecology*, vol. 48, no. 6, pp. 1432–1440, 2011.

[9] J. Senko, E. R. White, S. S. Heppell, and L. Gerber, "Comparing bycatch mitigation strategies for vulnerable marine megafauna," *Animal Conservation*, vol. 17, no. 1, pp. 5–18, 2014.

[10] A. Lucchetti, G. Bargione, A. Petetta, C. Vasapollo, and M. Virgili, "Reducing sea turtle bycatch in the mediterranean mixed demersal fisheries," *Frontiers in Marine Science*, vol. 6, p. 387, 2019.

[11] E. Gilman, J. Gearhart, B. Price, S. Eckert, H. Milliken, J. Wang, Y. Swimmer, D. Shiode, O. Abe, S. Hoyt Peckham *et al.*, "Mitigating sea turtle by-catch in coastal passive net fisheries," *Fish and Fisheries*, vol. 11, no. 1, pp. 57–88, 2010.

[12] J. H. Wang, S. Fisler, and Y. Swimmer, "Developing visual deterrents to reduce sea turtle bycatch in gill net fisheries," *Marine Ecology Progress Series*, vol. 408, pp. 241–250, 2010.

[13] M. Turk, A. Pentland, P. Belhumeur, and J. Hespanha, "Eigenfaces for recognition: Journal of cognitive neurosicence," 1991.

[14] Y.-H. Hsiao, C.-C. Chen, S.-I. Lin, and F.-P. Lin, "Real-world underwater fish recognition and identification, using sparse representation," *Ecological informatics*, vol. 23, pp. 13–21, 2014.

[15] Y. LeCun, F. J. Huang, and L. Bottou, "Learning methods for generic object recognition with invariance to pose and lighting," in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, vol. 2. IEEE, 2004, pp. II–104.

[16] H. S. Demir and E. Akagunduz, "Filter design for small target detection on infrared imagery using normalized-cross-correlation layer," *Turkish Journal of Electrical Engineering & Computer Sciences*, vol. 28, no. 1, pp. 302–317, 2020.

[17] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural computation*, vol. 1, no. 4, pp. 541–551, 1989.

[18] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *European conference on computer vision*. Springer, 2014, pp. 818–833.

[19] N. Zhang, J. Donahue, R. Girshick, and T. Darrell, "Part-based r-cnns for fine-grained category detection," in *European conference on computer vision*. Springer, 2014, pp. 834–849.

[20] X. Chen, S. Xiang, C.-L. Liu, and C.-H. Pan, "Vehicle detection in satellite images by hybrid deep convolutional neural networks," *IEEE Geoscience and remote sensing letters*, vol. 11, no. 10, pp. 1797–1801, 2014.

[21] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *CVPR09*, 2009.

[22] F. Storbeck and B. Daan, "Fish species recognition using computer vision and a neural network," *Fisheries Research*, vol. 51, no. 1, pp. 11–15, 2001.

[23] A. Salman, A. Jalal, F. Shafait, A. Mian, M. Shortis, J. Seager, and E. Harvey, "Fish species classification in unconstrained underwater environments based on deep learning," *Limnology and Oceanography: Methods*, vol. 14, no. 9, pp. 570–585, 2016.

[24] M.-C. Chuang, J.-N. Hwang, and K. Williams, "Supervised and unsupervised feature extraction methods for underwater fish species recognition," in *2014 ICPR Workshop on Computer Vision for Analysis of Underwater Imagery*. IEEE, 2014, pp. 33–40.

[25] G. Hu, K. Wang, Y. Peng, M. Qiu, J. Shi, and L. Liu, "Deep learning methods for underwater target feature extraction and recognition," *Computational intelligence and neuroscience*, vol. 2018, 2018.

[26] A. Purser, U. Hoge, J. Lemburg, Y. Bodur, E. Schiller, J. Ludszuweit, J. Greinert, and F. Wenzhöfer, "Plaspi marine cameras: Open-source, affordable camera systems for time series marine studies," *HardwareX*, p. e00102, 2020.

[27] A. Purser, Y. Marcon, S. Dreutter, U. Hoge, B. Sablotny, L. Hehemann, J. Lemburg, B. Dorschel, H. Biebow, and A. Boetius, "Ocean floor observation and bathymetry system (ofobs): a new towed camera/sonar system for deep-sea habitat surveys," *IEEE Journal of Oceanic Engineering*, vol. 44, no. 1, pp. 87–99, 2018.

[28] N. Strachan and L. Kell, "A potential method for the differentiation between haddock fish stocks by computer vision using canonical discriminant analysis," *ICES Journal of Marine Science*, vol. 52, no. 1, pp. 145–149, 1995.

[29] A. Rova, G. Mori, and L. M. Dill, "One fish, two fish, butterfish, trumpeter: Recognizing fish in underwater video." in *MVA*, 2007, pp. 404–407.

[30] C. Spampinato, D. Giordano, R. Di Salvo, Y.-H. J. Chen-Burger, R. B. Fisher, and G. Nadarajan, "Automatic fish classification for underwater species behavior understanding," in *Proceedings of the first ACM international workshop on Analysis and retrieval of tracked events and motion in imagery streams*, 2010, pp. 45–50.

[31] S. Mika, G. Ratsch, J. Weston, B. Scholkopf, and K.-R. Mullers, "Fisher discriminant analysis with kernels," in *Neural networks for signal processing IX: Proceedings of the 1999 IEEE signal processing society workshop (cat. no. 98th8468)*. Ieee, 1999, pp. 41–48.

[32] P. X. Huang, B. J. Boom, and R. B. Fisher, "Hierarchical classification with reject option for live fish recognition," *Machine Vision and Applications*, vol. 26, no. 1, pp. 89–102, 2015.

[33] K.-B. Duan and S. S. Keerthi, "Which is the best multiclass svm method? an empirical study," in *International workshop on multiple classifier systems*. Springer, 2005, pp. 278–285.

[34] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[35] M. A. Nielsen, *Neural networks and deep learning*. Determination press San Francisco, CA, USA:, 2015, vol. 2018.

[36] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[37] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[38] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2009.

[39] K. Kang, W. Ouyang, H. Li, and X. Wang, "Object detection from video tubelets with convolutional neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 817–825.

[40] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2009.

[41] H. Lu, Y. Li, L. Zhang, and S. Serikawa, "Contrast enhancement for images in turbid water," *JOSA A*, vol. 32, no. 5, pp. 886–893, 2015.

[42] R. Pi, "Raspberry pi 3 model b," *Online. Tillganglig: https://www.raspberrypi.org/products/raspberry-pi-2-model-b/Anvand 10 02 2016*, 2015.

[43] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *science*, vol. 220, no. 4598, pp. 671–680, 1983.

H. Seckin Demir received the B.S. and M.S. degrees in Electrical and Electronics Engineering from Bilkent University, Ankara, Turkey, in 2014 and 2017, respectively. Between 2014 and 2019, he worked as a computer vision scientist with ASELSAN Inc.

He is currently a PhD student and Graduate Research Assistant in Electrical, Computer and Energy Engineering (ECEE) Department at Arizona State University. His research interests include computer vision, image processing and machine learning.

Jennifer Blain Christen received a bachelor's degree (1999), master's degree (2001) and doctorate (2006) in electrical and computer engineering from Johns Hopkins University. Her dissertation focused on hybrid systems for life science applications exemplified through the development of a microincubator for cell culture. Blain Christen held a Graduate Research Fellowship and a G K-12 fellowship both from the National Science Foundation. In her postdoctoral work at the Johns Hopkins School of Medicine in the Immunogentics Department, she developed a microfluidic platform for homogeneous HLA (human leukocyte antigen) allele detection. Her research interests involve design of analog and mixed-mode integrated electronics for direct interface via innovative fabrication techniques to aqueous environments with special emphasis on biological materials.

Blain Christen is currently leading the BioElectrical Systems and Technology group at Arizona State University. The group has recently focused on point-of-care diagnostics and flexible neural interfaces. Point-of-care research involves sensing the presence of biomarkers in sweat and blood. They are funded by the National Science Foundation (NSF) to explore continuous sweat monitoring, and they are funded by the National Institutes of Health to create low-cost point of care HPV diagnostics for India. These projects in addition to other work in this area leverage Bluetooth communication (BLE) with smart phones to enable data transfer, storage and upload to the cloud. Blain Christen's work in flexible neural interfaces includes an NSF CAREER award exploring the use of optogenetics in the peripheral nervous system to enable future applications in bioelectronic medicine. She is also working with the Mayo Clinic to develop ultra-thin, high-resolution multi-modal neural arrays.

Blain Christen serves on the Board of Governors for the IEEE Circuits and Systems Society, and she is secretary of the Biomedical Circuits and Systems Technical Committee. She is also the advisor for the Medical 3D Printing student club.

Sule Ozev received her Ph.D. degree from University of California San Diego's Computer Science and Engineering Department in 2002. That same year, she joined Duke University's Electrical and Computer Engineering Department as an assistant professor. In 2008, she joined Arizona State University, School of Electrical, Computer and Energy Engineering, where she is currently a professor. She works on testing, calibration, and reliability of mixed-signal, radiofrequency, and sesnsor-based devices, built-in-self test techniques, analysis and mitigation of process variations, design and calibration of cyber-physical systems, and non-linear modeling of parametric relations with large dimensions. Her work is supported by U.S. National Science Foundation, Semiconductor Research Corporation, U.S. Department of Defense, and U.S. National Aeronautics and Space Administration. She has published more than 150 peer-reviewed conference and journal papers and holds two U.S. patents. Dr. Ozev also received 8 best paper awards at IEEE sponsored conferences.