Towards Learning-based Relative Underwater Pose Estimation Using a Single Light

Yuewei Fu
Department of Ocean Engineering
University of Rhode Island
Narragansett, RI, USA
yweifu@uri.edu

Mingxi Zhou
Graduate School of Oceanography
University of Rhode Island
Narragansett, RI, USA
mzhou@uri.edu

OCEANS 2024 - Halifax | 979-8-3315-4008-1/24/$31.00 ©2024 IEEE | DOI: 10.1109/OCEANS55160.2024.10754198

***Abstract*—Underwater localization is a challenging but an essential task for Autonomous Underwater Vehicles (AUV) operations. In this paper, we purpose utilizing learning-based methods to perform relative location estimation base on an active light source. This approach avoids the large latency associated with acoustic-based localization and the reduced detection range inherent in other vision-based techniques that rely on passive features or fiducial markers. Two styles of neural networks, a feedforward neural network (FNN) and a convolution neural network (CNN), with variation on the configurations, are implemented. Additionally, image processing pipeline for light features and feature concatenation of camera exposure time for CNN are discussed. Finally, the neural networks are evaluated on their capability of estimating relative location using a light in an indoor tank environment.**

I. Introduction

Accurate underwater localization is crucial for Autonomous Underwater Vehicles (AUV) operations and especially important in tasks such as docking that requires fast and precise prediction in complex and dynamics environments. Yet, it remains a challenging task as underwater is a GPS denial environment. Inertial sensors such as IMU and gyroscope combined with filtering techniques, can provide acceptable short-term localization results, but the error will growth unbounded over a long time horizon [1].

While acoustic baseline systems, such as the long baseline (LBL) or ultra-short baseline (USBL), provide good localization estimates at long range, they require additional setup [2]. Their acoustic properties also limits their operation frequency to few hertz [1], limiting the usefulness for localization at closed range.

Vision system paired with fiducial markers [3] has been used to achieve centimeter localization accuracy on ground robotics [4], however they are limited in underwater. For instance, the viewing distance of camera systems is greatly reduced and highly variable due to the nature of low ambient light and water clarity, resulting in a detectable distance of several meters in many scenarios [5].

On the other hand, active light markers have been explored for underwater localization [6], [7] due to their ability to overcome the issues with low visibility caused by low ambient light. However, previous works have been focused on either detecting the light on the image and control via lining up the light with the image center [6], or using multiple light markers with known baseline between the lights for pose estimation [7]. The first method of detecting light on the image is simple and straight forward but it has the drawback of only providing a relative location between the light and the vehicle in pixel coordinates, making its application limited to line-of-sight (LOS) style control. The second method of using multiple light markers converts the localization problem into a perspective-n-points (PnP) problem, making it capable of solving for pose in the physical world coordinates. However, to solve the PnP problem, it requires at at least 3 lights to be observed, which cannot be guaranteed when the AUV is in motion.

This paper explores the feasibility of a light-based relative localization using a single light source with a monocular camera. Relative local position [x, y, z] is estimated from a single image frame using two neural network configurations: a deep Feedforward Neural Network (FNN) and a Convolutional Neural Network (CNN). Both networks are trained on data we collected from an indoor tank.

In the remaining paper, we will present related works in Section II. The data collection, image processing, and neural network implementations are described in Section III. The evaluation of the neural networks are presented in Section IV. Lastly, conclusion and future work are discussed in Section V.

II. Background and Related Works

For short range localization, vision systems (cameras in particular), are frequently used in many area of field robotics due to their satisfying accuracy and accessibility from low cost of implementations. Many methods have been developed for vision based localization, and they can be separated into two groups: marker-based methods using fiducial landmarks placed in the scene, and marker-less approaches using feature detection and tracking techniques.

A. Marker Based Localization

Vision systems have been using fiducial to perform localization in tasks such as augmented reality with library such as

This work is supported by the National Science Foundation (NSF) under the award #2238168, and the Graduate School of Oceanography, University of Rhode Island.

979-8-3315-4008-1/24/$31.00 ©2024 IEEE

ARToolKit [8]. The development of AprilTag [9], a 2D square pattern that can be printed on any flat surface, has popularized this type of fiducial marker in the field of robotics. Its ease of use and robustness in detection, even with low-resolution cameras and partial occlusion of the marker, have made it particularly favored. Following this, newer version of AprilTag such as AprilTag 2 [10] and other variances of 2D planar style markers, such as ArUco [11], have been since developed and popularized with claims on enhanced performance.

These 2D planar style makers, such as the AprilTag, are typically monochromatic and are encoded in pattern of line that separate the tag into region white and black. These regions are then used by the detector to decode the identification information such as the tag family and number. Once detected, the tag's position and orientation can be determined in the camera frame by using the corners of the tag and solving the perspective-n-points (PnP) problem, which involves matching the corner pixels to their physical world coordinates. To enable this, the size of the tag and the intrinsic parameters of the camera need to be known.

AprilTag style fiducial markers have been used in applications of localization in underwater environment and have achieved various level of success [12]. However, their limitations are obvious: they require observable makers to be placed in the operating environment, which is not always permissible outside a pool environment, such as in the mid level of a water body, or when the operating field is large. An additional complexity in successfully implementing fiducial markers in underwater environments, compared to ground-based robotics, is the reduced visibility caused by large fluctuations in ambient lighting and increased water turbidity. This turbidity leads to light backscatter from suspended particles, significantly reducing the detection rate and limiting the detection range to only a few meters [5].

For application of underwater localization, active light emitting sources have been used in place of 2D planar style markers. In purposed methods of underwater docking for AUV system ([7], [13]–[16]), a common approach shared by these works are attaching lights to the docking station and detecting the light as a point in the image. With this setup, the localization problem can then be decomposed into finding a solution of the PnP problem. When the number of lights is greater or equal to 3, the relative position can be resolved. This active light approach, as opposite to AprilTag style marker, is favorable for the underwater environment as it effectively improve the usable range of the marker, making detection up around 15 meters possible as shown in [13].

A substantial downfall of the active light approach is when the number of observable light is less than 3, where the PnP problem is impossible to solve. This often occurs when the AUV is close to the light fixture (e.g. during entering), or when the AUV is viewing them from the side instead of directly facing towards them. Various works have used line-of-sight (LOS) method, where control goal is to align the light to the image center, instead for localization ([6], [17]). However, this method only provides the relative location of the light on the image in terms of pixels, and thus cannot be used to calculate physical distances. Consequently, it is primarily used for controlling the AUV rather than aiding in AUV localization.

B. *Marker-less Localization*

Marker-less localization are commonly used in field robotics with minimum modifications to the environments. They relies on the natural features present in the scene, such as edges and textures, or directly operate on the image features, such as intensities [18]. Techniques such as visual odometry [19] and Simultaneous Localization and Mapping (SLAM) are commonly employed and showed success in underwater environment such as seafloor mapping [20]. However, the performance of these systems heavily depend on the presents of features, and this is limits their usefulness for AUV systems beside those operating near the bottom of the sea.

Herein, we purpose a novel learning-based underwater localization scheme using a single light and monocular camera setup. To achieve pose estimation with a single light, we employ neural networks to learn the mapping between the photometric representation of the light in the image to the distance between the light and the camera.

III. Method

In this section, we first introduce the photometric image formation process, then we will discuss our data collection process, and present the network configurations.

A. *Photometric Image Formation*

The amount of radiance L inside the camera FOV from a light source can be formulated in Eq. 1, where Φ is the radiant flux emitted, $A{\cdot}cos\theta$ is the projected area normal to the incident flux, and ω is the solid angle [21]. The amount of light received at the camera sensor is called the Irradiance E, and it is not always equal to the radiance L due to the vignetting effect where intensities value drops toward the image boarder due to lens geometry. This can be modeled by the Cosine-Forth law [22], where Irradiance at sensor spatial location x to the light, can be modeled as the product of the vignetting function $V(x)$ and the radiance L as formulated in Eq. 2.

$$L = \frac{d^2\Phi}{dA \cdot cos\theta \cdot d\omega} \tag{1}$$

$$E(x) = V(x) \cdot L \tag{2}$$

$$E_{total}(x) = t \cdot E(x) \tag{3}$$

$$I(x) = G(t \cdot V(x) \cot L) \tag{4}$$

$$I(x) = G(t \cdot L(x)) \tag{5}$$

$$F(x) = f(G(t \cdot L(x))) \tag{6}$$

The camera sensor produce an image by opening the shutter for certain amount of time, which is called the exposure time t, and the total irradiance captured by the sensor is irradiance integrated over time, calculated following Eq. 3. The final pixel intensity, valued from 0 to 255 for an 8-bit image, is mapped from the total irradiance by the camera response function (CRF) $G : \mathbb{R} \rightarrow [0, 255]$ [23]. This is computed using Eq. 4.

In scenarios where the camera's dynamic range cannot balance the light from the light source against the background light, we can ignore the vignetting factor, as we are only interested in the light from the light source, which will saturate the pixel intensity. Then the image formation equation becomes Eq. 5, where L is a function of camera spatial location x because the radiant flux Φ and solid angle ω are constant and the only changes is in $A \cdot cos\theta$ which is a function of x. Then, the goal of the neural network is to learn the function (Eq. 6) which maps the pixel values, or some features extracted from the pixel values, to the camera location .

B. Data Collection

The instrument, camera, Apriltag markers, and light, used in data collection is shown in Fig. 1. A FLIR camera (packaged inside a 2 inch pressure housing with a flat view port) is used to collect images in an indoor water tank (7.5 m long, 4 m wide, 3.3 m deep). A Blue Robotics Lumen light is used as the source of lighting. Six Apriltag markers (3 on the left and 3 on the right of the light) are used for obtaining the ground truth position of the light to the camera. To obtain the ground truth relative location from the camera to the light, the relative locations of the AprilTag markers are estimated first. Then, a fixed transformation is applied to determine the relative location of the light.

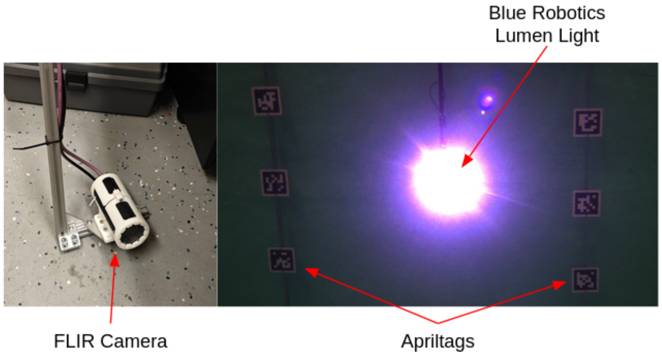

Fig. 1. The data collection instruments. Left: The FLIR camera used for data collection. Right: The configuration of Apriltag markers and the Lumen light for the Middleton tank data collection.

TABLE I
INDOOR TANK DATASET

Exposure Time (ms)	8	10	15	30	Total
Images	1848	770	565	123	3306

Before the collection process, a geometric calibration is performed on the camera using a checkerboard pattern in the tank. The focus on the FLIR camera is only manually adjustable, thus it is fixed for the rest of the data collection process after performing the calibration. During the collection process, the FLIR camera is configured at 20 fps, and a constant exposure time (range from 8 ms to 30 ms) is set during each run.

We placed the camera at nine locations in the indoor tank to view the light from different distances and angles. Every fourth frame from the image data is selected, and only images with at least four detected AprilTags are used to ensure robust ground truth pose estimation. Overall, a total of 3306 images are obtained for training and testing. The dataset consists of images with the distance to the light ranging from 2 m to 7.5 m. This range is limited by the size of the tank and the minimum distance to get the Apriltags in the field of view of the camera for obtaining the ground truth location.

Table I listed the numbers of images we obtained at different exposure time. We used 2214 images for model training, covering eight locations in the tank and all four exposure time settings, and the remaining 1092 images are used for testing, covering four locations with three exposure time settings (8 ms, 10 ms, 15 ms). The train and test sets are separated in this fashion to ensure that the test set consist only of images unseen during training, making over fitting to the test set less likely.

Algorithm 1: Shape Features Extraction

```
$I_{HSV} \leftarrow cvtColor(I_{RGB})$
/* Segment Light */
$(H,S,V)_{threshold}, (H,S,V)_{mask} \leftarrow Otsu(I_{H,S,V})$
$LC_{mask} \leftarrow S_{mask}$ & $V_{mask}$
$LR_{mask} \leftarrow H_{mask}$ & $V_{mask}$
/* Find Light Contours */
$LC_{contours} \leftarrow findContours(LC_{mask})$
$LR_{contours} \leftarrow findContours(LR_{mask})$
/* Find the correct contour */
for $contour \in LC_{contours}, LR_{contours}$ do
    /* Shape features */
    $Area_{ct}, Centroid_{ct}, Eccent_{ct}, \ldots$
    $Roundness_{ct} \leftarrow ShapeFeature(contour)$
    if $contour \in LC_{contours}$ then
        $i_{max\_area} \leftarrow argmax(Area_{ct})$
        /* Closest to image bottom */
        $i_y \leftarrow argmax(c_y)$
        if $i_{max\_area} = i_y$ then
            $LC_{contour} \leftarrow LC_{contours}[i_{max\_area}]$
            return $Area_{ct}, Centroid_{ct}, Eccent_{ct}$
                $Roundness_{ct}, (H,S,V)_{threshold}$
    if $contour \in LR_{contours}$ then
        $i_{max\_area} \leftarrow argmax(Area_{ct})$
        /* Closest to light center */
        $i_{distance} \leftarrow argmin(||Centroid_{ct} - LC_{ct}||)$
        if $i_{max\_area} = i_{distance}$ then
            $LR_{contour} \leftarrow LR_{contours}[i_{max\_area}]$
            return $Area_{ct}, Centroid_{ct}, Eccent_{ct}$
                $Roundness_{ct}, (H,S,V)_{threshold}$
```

C. Data Processing

Two deep neural networks are setup to compare the efficacy for pose estimation. We have configured a lightweight feedforward neural network (FNN) that requires pre-processing of the image data, and a more resource intensive convolution neural network (CNN) that is fully end-to-end. The architectures for each network is shown in Fig. 2. For CNN, the raw image is directly treated as the input. In contrast, we extracted several light features from the image as FNN inputs. For each image, two distinct features are extracted: light center and light ring. Light center is the bright saturated area at the light source,

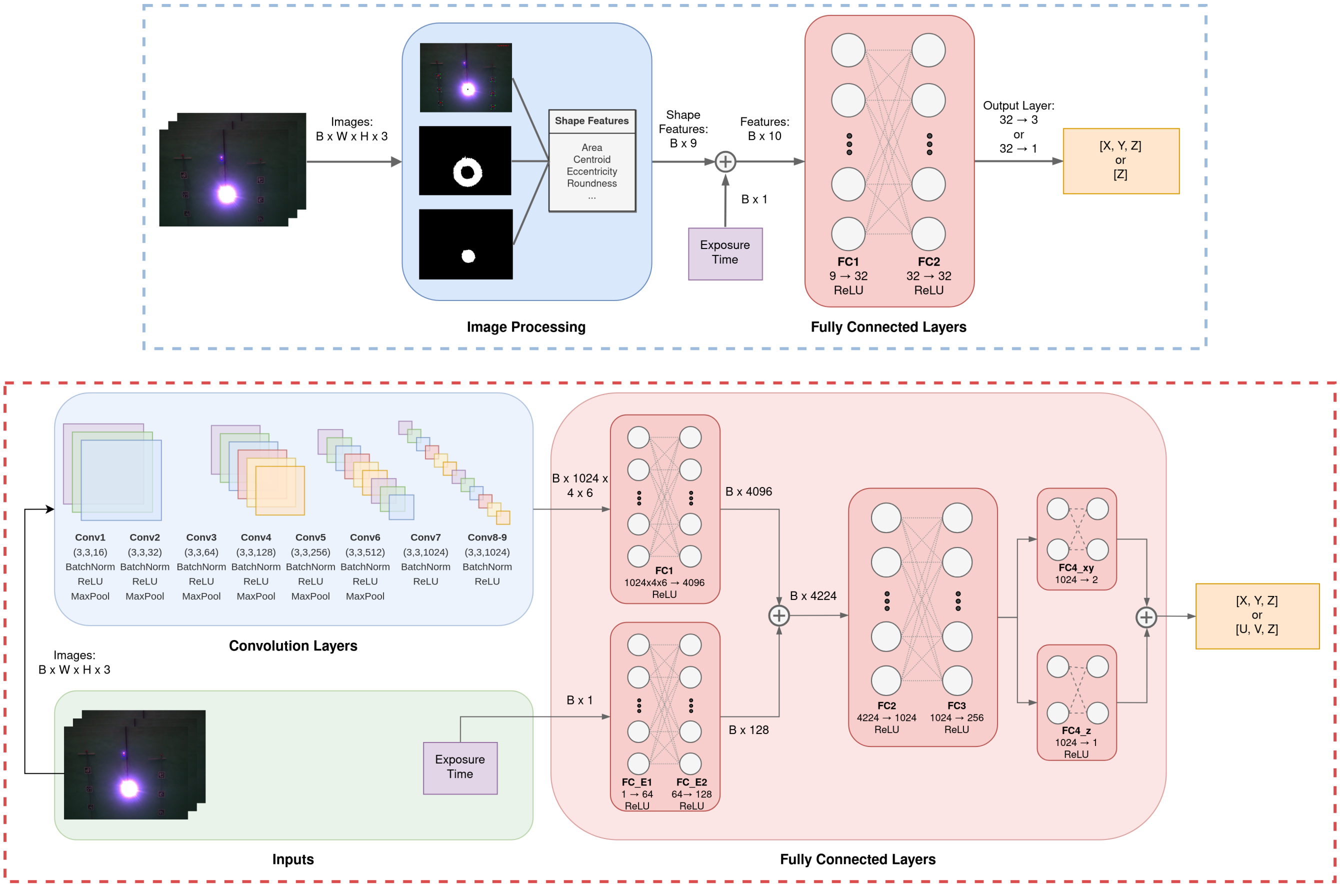

Fig. 2. The Top: FNN with two 32 neurons hidden layers. Bottom: CNN with 5 convolution layers and 3 fully connected layers.

while the light ring is outside the light center appeared as a abnormal ring of color that is most likely due to chromatic aberration.

In order to separate these two features out, we converted the raw RGB images into Hue (H), Saturation (S), Value (V) color space. Otsu's method [24], a method for maximizing the inter-class variance, is then applied. We applied Otsu's method to extract the light center mask (S and V channels) and light ring mask (H and V channels). The binary images are further analyzed and we have extracted ten features to be used as the input for FNN: the pixel location of the light center (P_x and P_y), the area and the roundness of the light center and light ring (A_{LC}, A_{LR}, $Roundness_{LC}$ and $Roundness_{LR}$), Otsu's thresholds ($H_{threshold}$, $S_{threshold}$, $V_{threshold}$), and the exposure time of the camera ($t_{exposure}$). The pseudo-code of the detailed procedure is shown in Algorithm 1 and 2.

D. Network Setups

Two configurations of feedforward neural network (FNN) with 2 hidden layers and 3 hidden layers each with 32 neurons are used. Ten features extracted from image processing are passed into the FNN. For each configuration, two variants of FNN are trained. The first one directly outputs the relative position [x, y, z] values, while the second outputs only predicts [z] and [x, y] are obtained using the re-projection equations in Eq. 7, where P_x, P_y are the pixel coordinates of the light center obtained from the shape features, c_x, c_x and f_x, f_y are camera principal point offsets and camera focal lengths obtained from camera calibration. Both variants of the FNN are trained for 100 epochs with a learning rate of 0.001.

Algorithm 2: Shape feature function

```
Function ShapeFeature($contour$):
    $M \leftarrow moments(contour)$
    /* Contour area */
    $Area_{ct} \leftarrow M[m00]$
    /* Contour centroid */
    $cx \leftarrow int(M[m10]/M[m00])$
    $cy \leftarrow int(M[m01]/M[m00])$
    $Centroid_{ct} \leftarrow (cx, cy)$
    /* Contour eccentricity */
    $mu_{20} \leftarrow M[mu20]$, $mu_{11} \leftarrow M[mu11]$
      $mu_{02} \leftarrow M[mu02]$ // 2nd order moments
    $cov \leftarrow [[mu_{20}, mu_{11}], [mu_{11}, mu_{02}]]$
    $eigval \leftarrow eigvals(cov)$
    $Eccent_{ct} \leftarrow \sqrt{(1 - min(eigval)/max(eigval))}$
    /* Contour Roundness */
    $hull \leftarrow convexHull(contour)$
    $perimeter \leftarrow arcLength(hull)$
    $Roundness_{ct} \leftarrow 4 * \pi * Area_{ct}/(perimeter^2)$
    return $Area_{ct}, Centroid_{ct}$
         $Eccent_{ct}, Roundness_{ct}$
```

TABLE II
MEAN ABSOLUTE ERROR (MAE) AND STANDARD DEVIATION OF ABSOLUTE ERROR (σ) OF X, Y, Z, AND EUCLIDEAN DISTANCE

Method	Output	MAE_x (m)	σ_x	MAE_y (m)	σ_y	MAE_z (m)	σ_z	MAE_ED (m)	σ_{ED}
FNN (32, 32)	x, y, z	0.21	0.14	0.11	0.09	0.47	0.36	0.56	0.34
FNN (32, 32, 32)	x, y, z	0.12	0.11	0.10	0.09	0.39	**0.22**	0.44	**0.22**
FNN (32, 32)	z	0.05	0.06	0.06	0.06	0.51	0.33	0.52	0.33
FNN (32, 32, 32)	z	**0.04**	**0.03**	**0.04**	**0.03**	**0.31**	0.23	**0.32**	0.23
CNN	x, y, z	0.17	0.15	0.21	0.28	0.74	0.43	0.83	0.47
CNN (t_exposure)	x, y, z	0.10	0.07	0.27	0.27	0.94	0.47	1.02	0.47
CNN	P_x, P_y, z	0.21	0.16	0.40	0.32	1.16	0.61	1.28	0.65
CNN (t_exposure)	P_x, P_y, z	0.21	0.14	0.22	0.30	1.03	0.53	1.13	0.51

$$x = \frac{z(P_x - c_x)}{f_x}, \quad y = \frac{z(P_y - c_y)}{f_y} \tag{7}$$

Two configurations of convolutional neural network (CNN) with two variation of outputs are also tested, with the base convolutional layers adapted from [15]. The first CNN directly outputs the relative position [x, y, z] values and the second CNN outputs [P_x, P_y, z] and the values of [x, y] are obtained by using Eq. 7. Two variations are tested for each CNN configuration, one with the input being just the image, and the second with exposure time as an additional input. The exposure time is passed through two layers of fully connected layers to convert into a feature vector of size 128. It is then concatenated with a feature vector of 4096 from the output of the convolution layers after passing through a fully connected layer. This balances the dimension to make the exposure time have similar influence for the subsequent layers.

The base convolutional module consists of 9 convolutional layers, with the first 6 layers performing a max pool operation to shrink the spatial dimension in half. All 9 layers uses a 3 by 3 kernel with a stride of 1 and padding of 1. BatchNorm operations are applied in each layer. Then, the output from the final convolutional layer is flatten and passed in to 4 fully connected layers. The CNN is a fully end-to-end network that require no manual processing of the data. All configurations and variants are trained for 100 epochs, with the [x, y, z] output configuration using a learning rate of 10^{-4} and the [P_x, P_y, z] configuration using a learning rate of 10^{-5}.

IV. RESULTS

The Mean Absolute Error (MAE) of x, y, z, and Euclidean distance (ED) are calculated for each network and are compared in Table II. Euclidean distance is the length of the line segment between the 3D coordinates of the light and the camera, and it can be calculated as shown in Eq. 8.

$$ED = \sqrt{(x_l - x_c)^2 + (y_l - y_c)^2 + (z_l - z_c)^2} \tag{8}$$

From Table II, it is evident that FNN networks generally outperformed CNN networks. The 3-hidden-layer FNN, which outputs z, performed the best, achieving the lowest MAE in [x, y, z, ED], with MAE_ED equal to 0.32 m

For the FNN network configurations, 3 hidden layers outperformed 2 hidden layers with the same output generally. The FNN variant that only outputs z has lower MAE_x and MAE_y as it directly uses the pixel value of the light center extracted in the light features to re-project back to the world coordinate [x, y], compared to counterparts that output [x, y, z] that has to learn the mapping of all light features to [x, y]. MAE_z is an important metrics as it directly compute the error in estimating the range of the light, and the 3 hidden layers predicted lower MAE_z compared to 2 hidden layers, meaning that it has more capability in learning the mapping function from light features to range.

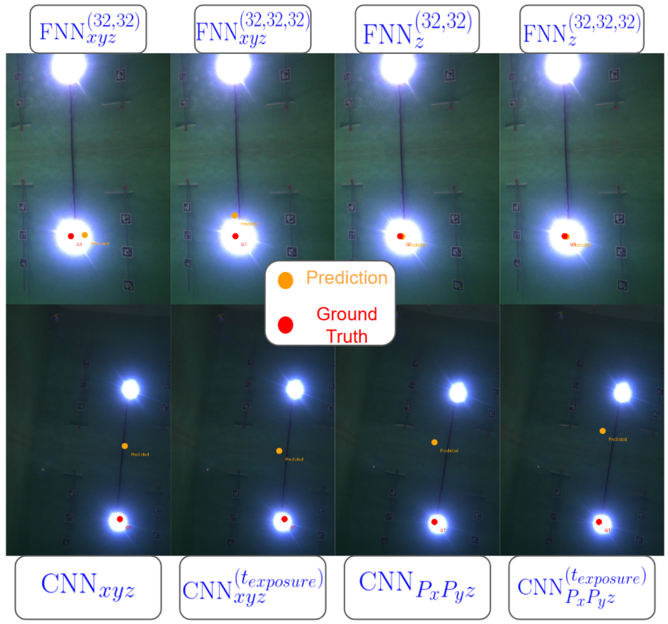

Fig. 3. The largest error in estimating the euclidean distance (ED) from each network is re-projected back onto the image (cropped). The prediction is labeled in orange and the ground truth is labeled in red.

For the CNN network configurations, the variants that output [x, y, z] performed better than the variants that output [P_x, P_y, z], indicating that training with the outputs in the same scale might allow the network to learn the mapping from pixel values to camera location relative to the light more effectively.

Reason for the worse performance of CNN networks compared to FNN can be seen in Fig. 3. The predictions of the test set with the worst MAE_ED are re-projected back to the image marked in orange and ground truth marked in red. A surface reflection of the light can be observed for all pictures, and the

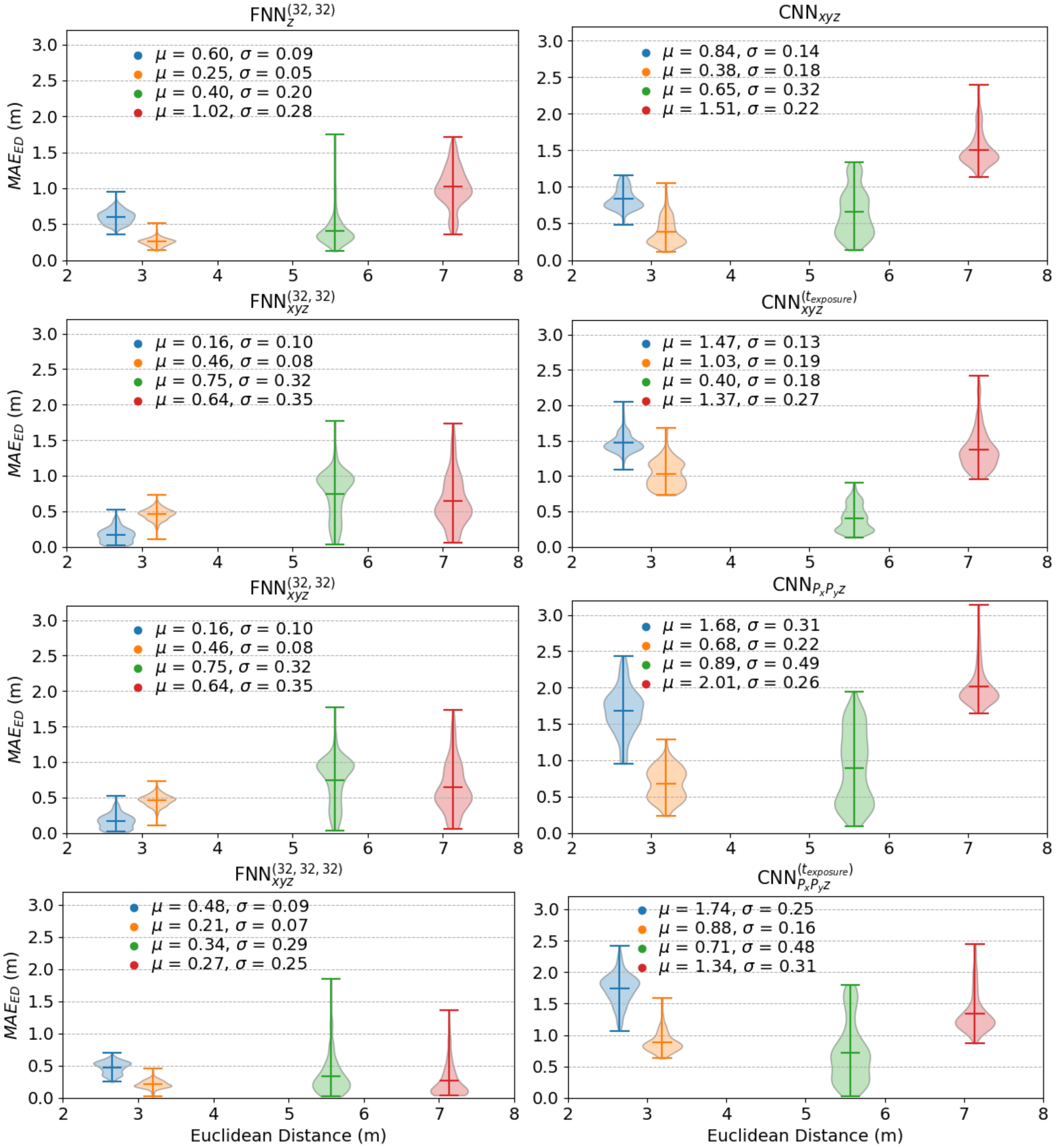

Fig. 4. The distribution of Mean Absolute Error (MAE) of Euclidean Distance (ED) clustered with ED are plotted for each networks.

CNNs tends to predict a location near the middle between the actual light and reflection. The FNNs have prediction centered near the actual light and unaffected by the surface reflection because the surface reflection is filtered out during image processing for the light features.

Furthuermore, the distributions of the MAE_{ED} from the test set prediction of each networks are plotted in Fig. 4. Each distribution is based on one location in the test, making a total of 4 distributions, and the distribution is plotted at the average euclidean distance of that particular location. It immediately obvious that for FNNs, the predictions have larger error extreme toward further away data. On the other hands, the CNNs produced large error even for closer distance data. It can also be observed for the additional exposure time input reduced the error for further away data but has the reverse impact for closer data.

V. Conclusion

In this paper, we proposed the use of learning-based methods for estimating the relative location of the camera using a single light. We have investigated two styles of neural networks, a feedforward neural network (FNN) and a convolution neural network (CNN), and tested with multiple configurations. Data was collected from an indoor tank at various locations and with various exposure time settings for the camera. First, we have demonstrated a pipeline for image processing, and using the feature extracted from the process, we have successfully trained 2 FNN configurations, each with 2 variants, and produced a lowest MAE_{ED} of 0.32 m. Second, we have implemented 2 CNN configurations, each with 2 variants, that is fully end-to-end without any manual processing or image processing. We have also constructed a method to include exposure time as a input to the CNN network. The lowest MAE_{ED} achieved with the CNN is 0.83 m. In future work, we plan to collect data from various water turbidity levels and utilize acoustic modems to extend the range at which we can obtain ground truth locations. This will help expand the effective range of our networks.

References

[1] L. Paull, S. Saeedi, M. Seto, and H. Li, "Auv navigation and localization: A review," *IEEE Journal of Oceanic Engineering*, vol. 39, no. 1, pp. 131–149, 2014.

[2] N. R. Rypkema and H. Schmidt, "Passive inverted ultra-short baseline (piusbl) localization: An experimental evaluation of accuracy," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 7197–7204.

[3] E. Olson, "Apriltag: A robust and flexible visual fiducial system," in *2011 IEEE International Conference on Robotics and Automation*, 2011, pp. 3400–3407.

[4] M. Kalaitzakis, S. Carroll, A. Ambrosi, C. Whitehead, and N. Vitzilaios, "Experimental comparison of fiducial markers for pose estimation," in *2020 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2020, pp. 781–789.

[5] D. B. dos Santos Cesar, C. Gaudig, M. Fritsche, M. A. dos Reis, and F. Kirchner, "An evaluation of artificial fiducial markers in underwater environments," in *OCEANS 2015 - Genova*, 2015, pp. 1–6.

[6] J. Chavez-Galaviz and N. Mahmoudian, "Underwater dock detection through convolutional neural networks trained with artificial image generation," in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 4621–4627.

[7] N. Palomeras *et al.*, "Auv homing and docking for remote operations," *Ocean Engineering*, vol. 154, pp. 106–120, 2018.

[8] D. Wagner, G. Reitmayr, A. Mulloni, T. Drummond, and D. Schmalstieg, "Pose tracking from natural features on mobile phones," in *2008 7th IEEE/ACM International Symposium on Mixed and Augmented Reality*, 2008, pp. 125–134.

[9] E. Olson, "Apriltag: A robust and flexible visual fiducial system," in *2011 IEEE International Conference on Robotics and Automation*, 2011, pp. 3400–3407.

[10] J. Wang and E. Olson, "Apriltag 2: Efficient and robust fiducial detection," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 4193–4198.

[11] S. Garrido-Jurado, R. Muñoz-Salinas, F. Madrid-Cuevas, and M. Marín-Jiménez, "Automatic generation and detection of highly reliable fiducial markers under occlusion," *Pattern Recognition*, vol. 47, no. 6, pp. 2280–2292, 2014.

[12] J. Chen, C. Sun, and A. Zhang, "Autonomous navigation for adaptive unmanned underwater vehicles using fiducial markers," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 9298–9304.

[13] S. Xu *et al.*, "A stereo visual navigation method for docking autonomous underwater vehicles," *Journal of Field Robotics*, vol. 41, no. 2, pp. 374–395, 2024.

[14] Z. Zhang, L. Zhong, M. Lin, R. Lin, and D. Li, "Triangle codes and tracer lights based absolute positioning method for terminal visual docking of autonomous underwater vehicles," *Industrial Robot*, vol. 51, no. 2, pp. 269–286, 2024.

[15] S. Liu, M. Ozay, T. Okatani, H. Xu, K. Sun, and Y. Lin, "Detection and pose estimation for short-range vision-based underwater docking," *IEEE Access*, vol. 7, pp. 2720–2749, 2019.

[16] P. Trslić, A. Weir, J. Riordan, E. Omerdic, D. Toal, and G. Dooly, "Vision-based localization system suited to resident underwater vehicles," *Sensors*, vol. 20, no. 2, 2020.

[17] M. Lin *et al.*, "Docking to an underwater suspended charging station: Systematic design and experimental tests," *Ocean Engineering*, vol. 249, p. 110766, 2022.

[18] Y. Wu, X. Ta, R. Xiao, Y. Wei, D. An, and D. Li, "Survey of underwater robot positioning navigation," *Applied Ocean Research*, vol. 90, p. 101845, 2019.

[19] D. Nister, O. Naroditsky, and J. Bergen, "Visual odometry," in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, vol. 1, 2004, pp. I–I.

[20] S. Zhang *et al.*, "Visual slam for underwater vehicles: A survey," *Computer Science Review*, vol. 46, p. 100510, 2022.

[21] F. E. Nicodemus, J. C. Richmond, J. J. Hsia, I. W. Ginsberg, and T. Limperis, *Geometrical considerations and nomenclature for reflectance*. USA: Jones and Bartlett Publishers, Inc., 1992, p. 94–145.

[22] S. J. Kim and M. Pollefeys, "Robust radiometric calibration and vignetting correction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 4, pp. 562–576, 2008.

[23] P. Bergmann, R. Wang, and D. Cremers, "Online photometric calibration of auto exposure video for realtime visual odometry and slam," *IEEE Robotics and Automation Letters*, vol. 3, pp. 627–634, 2017. [Online]. Available: https://api.semanticscholar.org/CorpusID:3676655

[24] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.