***In* CIBB 2021: Computational Intelligence Methods for Bioinformatics and Biostatistics proceedings. Part of the Lecture Notes in Computer Science book series (LNBI,volume 13483). pp 25–39**
2022
Springer, Cham.
https://doi.org/10.1007/978-3-031-20837-9_3
https://archimer.ifremer.fr/doc/00807/91900/

Real-Time Automatic Plankton Detection, Tracking and Classification on Raw Hologram

Scherrer Romane [1],, * Govan Rodrigue [1], Quiniou Thomas [1], Jauffrais Thierry [2], Lemonnier Hugues [2], Bonnet Sophie [3], Selmaoui-Folcher Nazha [1]

[1] ISEA, Universite de la Nouvelle-Cal´edonie, Noumea, New-Caledonia
[2] Ifremer, UMR9220 Entropie, Noumea, New-Caledonia.
[3] Aix Marseille Univ, Universit´e de Toulon, CNRS, IRD, MIO, Marseille, France

* Corresponding author : Romane Scherrer, email address : romane.scherrer@hotmail.fr

Abstract :

Digital holography is an imaging process that encodes the 3D information of objects into a single intensity image. In recent years, this technology has been used to detect and count various microscopic objects and has been applied in submersible equipment to monitor the in situ distribution of plankton. To count and classify plankton, conventional methods require a holographic reconstruction step to decode the hologram before identifying the objects. However, this iterative and time-consuming step must be performed at each frame of a video, which makes it difficult to support real-time processing. We propose a real-time object detection based approach that simultaneously performs the detection, classification and counting of all plankton within videos of raw holograms. Experiments show that our pipeline based on YOLOv5 and SORT is fast (44 FPS) and can accurately detect and identify the plankton among 13 classes (97.6% mAP@0.5, 92% MOTA). Our method can be implemented to detect and count other microscopic objects in raw holograms.

Keywords : Object detection, Multiple Object Tracking, Deep learning, Plankton, Digital holography

1 Introduction

The observation, counts and classification of marine plankton are essential to measure the health of our oceans. In recent years, several submersible equipment [8] (ISIIS, LISST-Holo, eHoloCam) have been deployed as part of large-scale campaigns to acquire *in situ* images of plankton. Some of these systems use digital holography [14], a method that enable high resolution images acquisition over a large water column and at high flow rates. Since a hologram encodes the 3D information of all plankton as a single intensity image, a decoding process, called holographic reconstruction, is required to retrieve the sample image from its hologram. Unfortunately, the methods used to process holograms and then count and classify the species are still very time-consuming and manual.

With the multiplication of collected images, various efforts have been made to accelerate and improve the holographic reconstruction, for instance, by adopting

a convolutional neural network (CNN) to automatically find the focus [18] or to reconstruct a de-focused hologram without performing an auto-focusing or phase recovery routine [16, 21]. Even though those approaches greatly accelerate the holographic reconstruction, the detection and classification of the objects need to be performed afterwards.

To count and identify the objects in a live video stream, three different tasks are necessary: (i) a classification task to identify the objects, (ii) a detection task to locate them and (iii) a tracking task to determine their respective trajectories to avoid counting the same objects several time during the video life span. However, these three distinct, yet complementary, tasks are often performed independently on holograms. The classification is often done on cropped holograms with, for example, a trained CNN as in [4, 22] but a preliminary detection is necessary to determine those regions of interest (ROIs) that are then feed into the model. To detect the objects, some works have implemented a CNN-based sliding window algorithm [19] that perform a binary classification on different regions in the holograms to detect and count cells. Other studies propose to perform the detection with a segmentation-based method. The segmentation can be carried out with a threshold as in [17] that proposes to filter the intensity of the reconstructed holograms with a bandpass filter before applying a threshold to generate a binary mask. The segmentation can also be done with a deep learning model as in [7] where a Segnet model coupled with a circular Hough transform are applied on the holograms to locate the objects. However, detection by segmentation often requires a prior holographic reconstruction, as the diffraction patterns on raw holograms do not allow the object's boundaries to be precisely determined. Concerning the tracking task, which is performed to determine the objects trajectories, the existing methods are generally based on a frame-by-frame detection of the objects that are then associated through the sequences [10]. In the framework of holography, the detection assignment can be carried out with the calculation of the cross-correlation between two consecutive frames [13] which is effective when there is little variation in object morphology or noise between the images. When the motion of the objects causes a variation of their morphology (spin, rotation) between frames, other more robust algorithms, such as the minimum boundary filter (MBF) [9], have been successfully applied. However, these methods rely on a detection pipeline that requires a holographic reconstruction at each frame of the video.

Even if several approaches have been proposed in the last few years to detect and classify objects on holograms, the methods often focus on only one aspect, either a classification or a detection/tracking task. Moreover, most of the existing methods require a prior holographic reconstruction to detect the objects [20]. However, conventional algorithms [5] used to search each object's focus plane and remove twin image artifacts are iterative and computationally intensive and therefore not always compatible with real-time processing. Therefore, the use of an object detection model such as [12] Faster-RCNN, YOLO, SSD or RefineDet, offers an alternative by performing in real time the localization and classification of all objects on a frame in a single pass. Applied to raw holograms, these real-

time models could greatly improve the applicability of digital holography and are compatible with other tracking algorithm to accurately count and classify the objects.

The aim of the paper is to demonstrate that the classification, localization and tracking of plankton can be simultaneously performed in real-time on raw holograms with an object detection model. For that purpose, two datasets of labeled in-line holograms will be simulated with 13 different plankton species. The paper is organized as follows. In the next section, the generation of holographic datasets and the object detection models are described. Section three shows the performances of the models. Conclusions and perspectives are given in the last section.

2 Materials and Methods

2.1 Hologram Formation

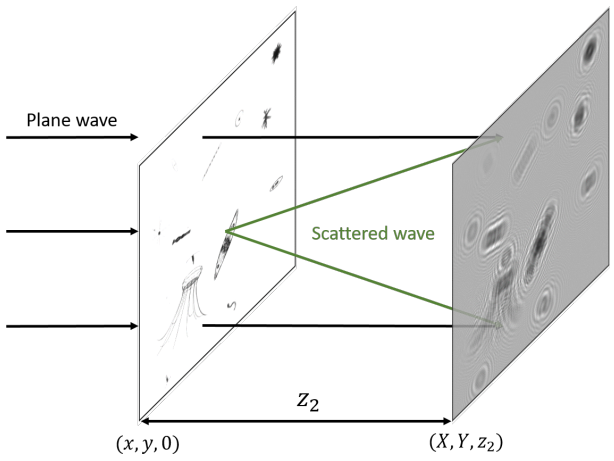

Fig. 1. In line-holography.

For an in-line holographic setup (see Fig. 1), the reference and object waves share the same optical axis and an object can be described by a complex transmission function [6] at a given z plane:

$$t_z(x, y) = \exp[-a(x, y)] \exp[i\phi(x, y)] \tag{1}$$

where $a(x, y)$ describes the absorption of the object and $\phi(x, y)$ is the phase distribution. The transmission function can be used to calculate the wavefront just behind the object $U_{z^+}(x, y)$:

$$U_{z^+}(x, y) = t_z(x, y) U_{z^-}(x, y) \tag{2}$$

where $U_{z^-}(x, y)$ is the incident wave that can be either plane or spherical. Considering that the object is located at $z = 0$, the exit wave given by Eq. 2 can be rewritten as $U_{0^+}(x, y) = t_0(x, u)U_{0^-}(x, y)$ and is propagated to the detector/hologram plane which is located at $z = z_2$ along the optical axis. This propagation is simulated by the angular spectrum method by calculation of the following transformation:

$$U_{z_2}(X, Y) = TF^{-1}\left[TF\left(U_{0^+}(x, y)\right) \times \exp\left(\frac{2\pi i z_2}{\lambda}\sqrt{1 - (\lambda u)^2 - (\lambda v)^2}\right)\right] \quad (3)$$

where λ is the wavelength and (u, v) are the Fourier domain coordinates. TF^{-1} and TF denoted the inverse and the direct Fourier transform, respectively. Note that Eq. 3 is often expressed as $U_{z_2}(X, Y) = R(X, Y) + O(X, Y)$ where R and O are the reference and the object waves that interfere at the surface of the recording medium. The recorded hologram at $z = z_2$ is the intensity calculated by:

$$H_{z_2}(X, Y) = |U_{z_2}(X, Y)|^2 = U_{z_2}(X, Y)U^*_{z_2}(X, Y) \quad (4)$$

where * denotes the complex conjugate. As a result, a hologram can be simulated once λ, z_2, $U_{0^-}(x, y)$ and $t_0(x, y)$ are known or set.

2.2 Dataset

Plankton Images To generate a dataset of labeled holograms for an object detection task, the complex transmission function $t_0(x, y)$ of several objects in a plane $(x, y, z = 0)$ must be simulated first. For that purpose, two labeled datasets of plankton images will be used as objects. The first dataset consists of shadow images collected by the In Situ Ichthyoplankton Imaging System (ISIIS), which was the subject of a competition on Kaggle [4]. This open source dataset consists of 121 marine plankton species, among which 10 species with a number of images greater than 1000 were selected for our simulations. The second dataset (custom) consists of optical microscopy images of 3 phyto-plankton species from New Caledonia (*Haslea sp., Pleurosigma sp.* and *Mastogloia sp.* noted P1, P16 and P17, respectively). The plankton was imaged with a bright-field microscope at a $\times 10$ magnification. The images were automatically thresholded, segmented into ROIs using an edge detection based algorithm (Sobel) and manually labeled. Fig. 2 presents the number of images per species. Note that for each dataset, the ROI segments are labeled per class and saved as grayscale images. Moreover, the images were processed so that background has a constant value equal to 1 and only the pixels inside the object support have a value between 0 and 1. This particularly allow us to simulate the absorption $a(x, y)$ and the transmission function $t(x, y)$ of the objects. In particular, we converted a ROI segment $I(x, y)$ into an absorption with $a(x, y) = -1 \times I(x, y) + 1$ so that the transmission function is $t_0(x, y) = \exp[-a(x, y)] \exp[i\phi(x, y)]$ inside the object support and $t_0(x, y) = 1$ where there is no object ($a(x, y) = 0$ and $\phi(x, y) = 0$). Note that

[4] https://www.kaggle.com/c/datasciencebowl/

$t_0(x, y) = 1$ only implies that the incident wave that illuminates the sample remains undisturbed where there is no plankton ($U_0^+(x, y) = U_0^-(x, y)$).

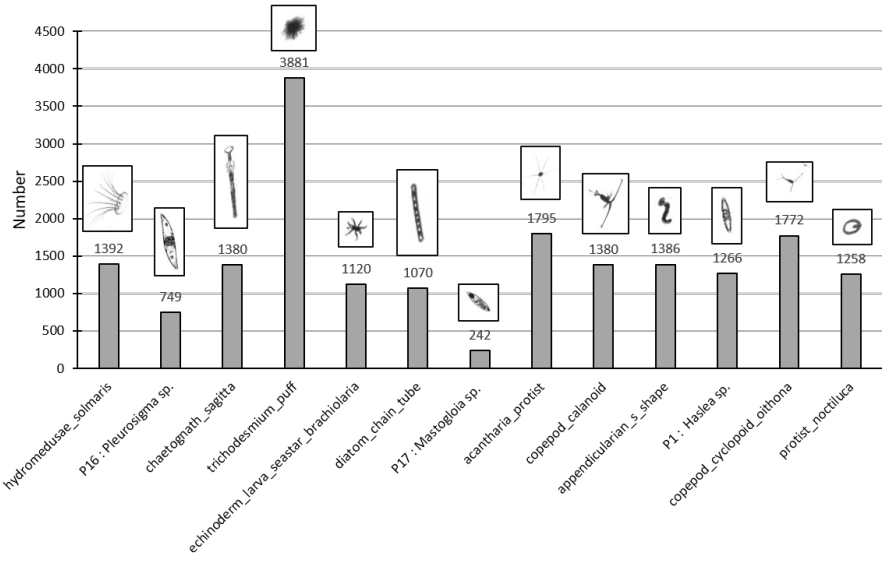

Fig. 2. Number of images per species.

To simulate $t_0(x, y)$ with various objects, the transmission functions of several plankton images can be randomly placed on a $N \times N$ empty (all-ones) image. By doing so, the (x, y)-axis coordinates of the bounding boxes are randomly set. Moreover, the plankton images are already saved as ROIs so that the bounding boxes width and height are the images dimensions. Since the images are classified per species, the labels of a simulated $t_0(x, y)$ for an object detection task (classes and bounding boxes coordinates) can be completely set. Once $t_0(x, y)$ is simulated, the corresponding hologram can be computed with the Eq. 3 and Eq. 4.

Holograms Simulation To demonstrate that it is possible to classify and track objects on raw holograms in real time, we have simulated two datasets. The first dataset, used to train and test the detection model, consists of 10,000 simulated holograms. The second dataset, used to evaluate the tracking performance of the model, is composed of 100 simulated videos in which plankton are moving in a laminar flow in a two-dimensional plane channel. In this section, we describe in more detail the simulation of these two datasets.

Object detection dataset Before simulating the transmission functions and the corresponding holograms to train the detection model, the plankton images from

the two sample image datasets (ISIIS and Custom) were randomly split, per class, in a 80:20 ratio for training and testing, respectively. We have considered that the plankton are pure amplitude objects so that $\phi(x, y) = 0$. The simulation of $t_0(x, y)$ proceeds as follows. First, for each simulated $t_0(x, y)$, 13 plankton images (one per species) are randomly selected. The images are then randomly rotated and flipped with 4 possible rotations (0°,90°,180° or 270°) and 3 possible flips (None, horizontal or vertical). Then, the plankton transmission functions are individually modified so that $t_{plankton}(x, y) = \exp[-C \times a(x, y)]$ where C is a random constant and $C \in [0.5, 1]$. Next, the 13 transmission functions are randomly placed without overlapping on a 512×512 empty image to generate $t_0(x, y)$. Finally, the hologram $H_{z2}(X, Y)$ is simulated with Eq. 3 and Eq. 4. Both the holograms and the $t_0(x, y)$ are normalized between 0 and 1 and saved. 8,000 and 2,000 holograms were simulated for training and testing, respectively. Fig. 3 presents an example of a simulated and labeled $t_0(x, y)$ and its corresponding hologram. During training, the object detection model learns to locate all the plankton on the raw holograms. The model should be able to predict the bounding boxes of the objects (x,y,w,h) and the class.

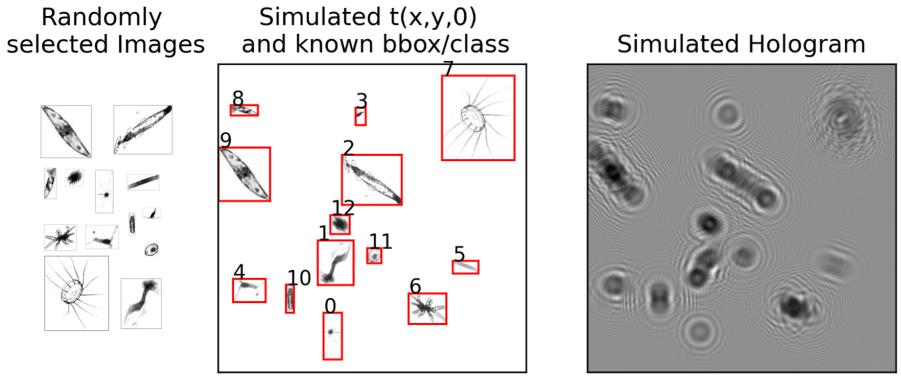

Fig. 3. Simulation example. 13 plankton images are used to generate a labeled hologram for a object detection task.

Tracking dataset To evaluate the tracking performances, we have simulated 100 videos that consist of 50 frames in which several plankton are moving in a 2D channel. For each simulated video, 10 plankton images were randomly selected from the ROIs used to test the detection model. For each selected plankton, we have simulated the transmission function $t_{plankton}(x, y) = \exp[-C \times a(x, y)], C \in [0.5, 1]$ which remained constant throughout the video. The plankton was then randomly placed on a 512×512 all-ones image with a non-overlapping constraint, so that the plankton does not initially occluded a previously placed plankton. Its velocity was then initialized with the calculation of the Poiseuille equation

between two planes. Note that, for each video that lasts 50 frames, the 10 plankton are appearing or disappearing at different frame index according to their respective speed and frame of appearance (see Fig. 4).

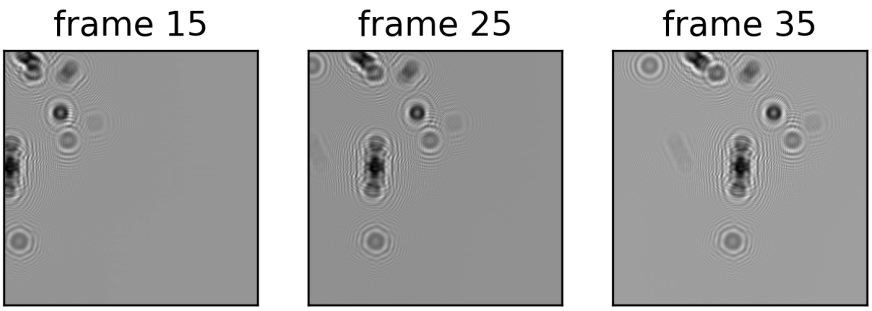

Fig. 4. Tracking dataset example. 10 plankton are moving in a 2D channel.

For the simulations of the two datasets, we have considered $\lambda = 530$ nm (green), $z_2 = 0.8$ mm and a pixel size, which limits the final resolution, of 1.12 μm. The incident plane wave, described by a distribution $U(z) = \exp[i(k_x x + k_y y + k_z z)]$ where (k_x, k_y, k_z) are the wave vector components, was simulated with $U_0^-(x, y) = 1$ by choosing the position of the object at $z = 0$ and by selecting the optical axis along the propagation of the wave ($k_x = k_y = 0$) [6]. Note that since the plankton are placed close to the camera plane ($z_2 <$ 1 mm), the simulated holograms are captured with a unit magnification [11]. As a result, the models trained on 512x512 images should be able to detect plankton over a field of view equal to 0.33 mm^2. The source code is available at https://github.com/romanescherrer/HoloTrack.

2.3 Object Detection Models and Tracking

Two tasks are to be considered in this paper. The first is the detection of objects on raw holograms which is performed frame by frame on a video. By detection, we mean the localization of all objects i.e. the determination of bounding boxes of coordinates (x,y,w,h) and the classification of objects (one among the considered 13 classes). The second task is the tracking of the objects throughout the video. This task, which aims at associating/linking detections across frames, allows, among other things, to determine the objects trajectories in order to precisely count the plankton that appear and disappear in the video without generating any duplicate.

Detection To perform object detection task on raw holograms, we chose two YOLOv5 [3, 15] models that were pre-trained on the COCO dataset, namely

YOLOv5s [5] (the smallest) and YOLOv5x (the largest) with $7.3M$ and $87.7M$ parameters, respectively. YOLO is a one-stage detector that integrates the detection of objects and their respective classification into a single process and has achieved state-of-the-art performances in term of speed and accuracy in many object detection problems. The model is composed of 3 parts (Fig. 5): a backbone (CSPDarknet), a neck (PANet) and a head (Yolo) that collect features from different stages of a $N \times N$ input images and encode/decode them into 3 output tensors of size $S \times S \times (B * (5 + n_c))$ where $S = (N/32, N/16, N/8)$, B is the number of anchors per grid cell and n_c is the number of classes. The anchors are generic bounding boxes dimensions (w,h) that are determined using a clustering algorithm (k-means) on the training dataset. Each cell in an output tensor is responsible for detecting objects within itself and after various post-processing steps (non-max suppression, among other, to only retain the candidate bounding boxes with higher response [3]), YOLO produces an output prediction vector $p = (b, o, c)$ where $b = (x, y, w, h)$ are the objects bounding boxes, o is the objectness i.e. a confidence score that the bounding boxes captures real objects and c is the class of the objects.

The models were trained on 8,000 holograms during 400 epochs with a batch size of 8 and tested on 2,000 holograms. The SGD optimizer was used with an initial learning rate equal to 0.01. To further evaluate the object detection performances on raw holograms, two models were also trained on the transmission functions $t_0(x, u)$ with the same hyperparameters. Note that $t_0(x, y)$ is the perfect image (artifact-free) that the holographic reconstruction steps seeks to obtain. Comparing the detection results on the holograms with those obtained on transmission function allow to determine whether the holographic reconstruction steps, which are iterative and time consuming, are avoidable to accurately classify and locate the objects with precision. The experiments were carried out on a 2.9 GHz Intel Core i7 PC with 64 GB of RAM and a Nvidia GTX 2060 GPU. The training took 8 hours for the small model and 2 days for the larger one.

Tracking Yolo is a real-time object detector [3] and thus can predict the bounding boxes and the classes of the objects at every frame of the video. In order to associate/link the detections across frames, we used the SORT algorithm proposed by [2]. The method works as follows (Fig. 6): During the algorithm initialization at the first frame noted k, each bounding box d_k detected by YOLO is associated with an unique tracker which is composed of a kalman filter. We denote $t_k \# n$ the bounding boxes of the trackers at the frame k where n is an unique identifier. For the next frame $k + 1$, the new bounding boxes d_{k+1} detected by YOLO must be associated to the existing trackers or new trackers must be created if the objects were not detected at the previous frame. For this, the kalman filters of the trackers predict the state of the bounding boxes at frame $k + 1$ by knowing the state of the bounding boxes at frame k. Then, the association of $d_{k+1} \# m, m \in [1, 2, ..., M]$ with the bounding boxes of

[5] https://github.com/ultralytics/yolov5

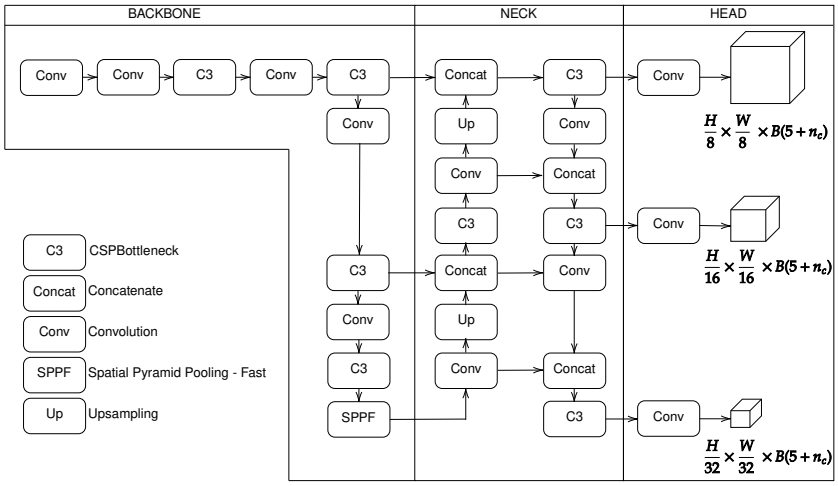

Fig. 5. YOLOv5 architecture.

the trackers $t_{k+1}\#n, n \in [1, 2, ...N]$ is performed by computing a cost matrix $C = \big(- IoU(d_{k+1}\#m, t_{k+1}\#n)\big) \in \mathbb{R}^{M \times N}$ where IoU is the Intersection over Union expressed by :

$$IoU(d_{k+1}, t_{k+1}) = \frac{d_{k+1} \cap t_{k+1}}{d_{k+1} \cup t_{k+1}} \tag{5}$$

The assignment is solved using the Hungarian algorithm and once a detection is associated to a target, the detected bounding box d_{k+1} is used to update the target state via the associated Kalman filter. The SORT algorithm is applied sequentially, frame by frame after the YOLO inference, on the whole video stream and the tracking can be done in real time because the state of the system at frame k is predicted by its previous state at frame $k-1$.

2.4 Metrics

To evaluate YOLO, we report the object detection performances with the well-known average precision (AP) metrics [12]. We recall that the AP@.5 and AP@.75 are the average precision computed with an intersection over union threshold $t = 0.5$ and $t = 0.75$, respectively. The AP@[.5:.95] is reported by computing the mean AP@ with 10 different IoU thresholds [.5:.05:.95].
To evaluate the tracking performances, we report the CLEAR MOT metrics [1], with in particular:

– **MOTA**: The Multiple Object Tracking Accuracy metric that combines the false negative rate (FN), false positive rate (FP) and the mismatch rate ($IDSW$) into a single score :

$$MOTA = 1 - \frac{\sum_t (FN_t + FP_t + IDSW_t)}{\sum_t GT_t} \tag{6}$$

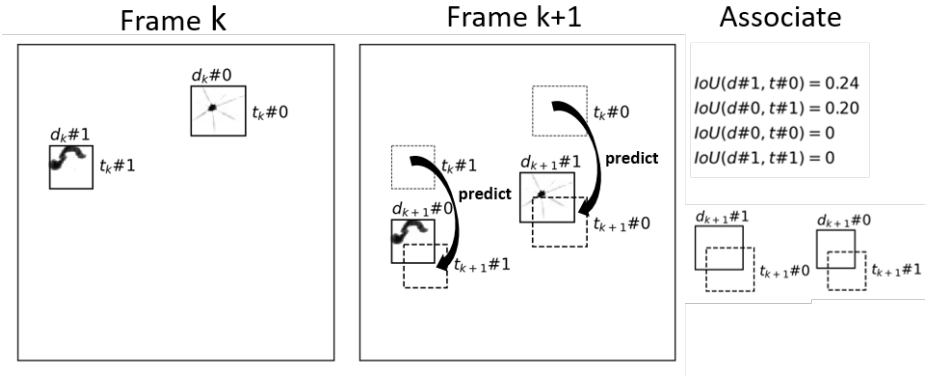

Fig. 6. SORT multiple object detection algorithm.

where t is the frame index and GT is the number of ground-truth objects.

- **MOTP**: The Multiple Object Tracking Precision that describes how precisely the objects are tracked by measuring and averaging the IoU between the objects and their corresponding hypothesis.

$$MOTP = \frac{\sum_{t,i} d_{t,i}}{\sum_t c_t} \tag{7}$$

where $d_{t,i}$ is the bounding boxes overlap between the target i and its assigned ground-truth objet and c_t is the number of matches.

- **MTR**: The Mostly Tracked Rate which is the percentage of ground-truth tracks that have the same label for at least 80 % of their life span.
- **MLR**: The Mostly Lost Rate which is the percentage of ground-truth tracks that are tracked for less that 20 % of their life span.

3 Results

3.1 Detection performances

In this section, we report the object detection performances on 2000 test holograms and the mean inference time that includes FP16 inference, postprocessing and non-max suppression on a GTX 2060 GPU. Tab. 1 summarizes the performances of the object detection tasks performed on the raw holograms and on the transmission functions $t_0(x, y)$.

For the models trained on the holograms, the AP@.5 are 0.976 and 0.981 for YOLOv5s and YOLOv5x, respectively. For the models trained on the transmission functions, the AP@.5 are slightly better with 0.985 and 0.993 for YOLOv5s and YOLOv5x, respectively. The AP@[.5:.95] are significantly higher on $t_0(x, y)$ than on holograms (eg. 0.980 vs. 0.855 for YOLOv5x) but the AP@.75 are still high on holograms (0.928 and 0.955 for YOLOv5s and YOLOv5x, respectively).

Table 1. Detection Performances.

Model	Inputs	AP@.5:.95	AP@.5	AP@.75	Speed
YOLOv5s	Holograms	0.820	0.976	0.928	4 ms
	$t_0(x, y)$	0.967	0.985	0.985	
YOLOv5x	Holograms	0.855	0.981	0.955	14 ms
	$t_0(x, y)$	0.980	0.993	0.989	

Those results suggest that the detectors trained on the holograms are efficient for a IoU threshold $\leq .75$ but that their performances start to decline at a higher threshold. Fig. 7 shows the confusion matrix of YOLOv5x at IoU@.5 on the test holograms and an example of its predictions. One can notice that the diffraction pattern of an object spreads beyond its bounding box. In fact, the further away the object is from the camera, the more this effect will be visible on the hologram. Because of this and the lack of sharp edge, a detector trained on holograms was expected to have difficulty in determining the object boundaries with a high IoU.

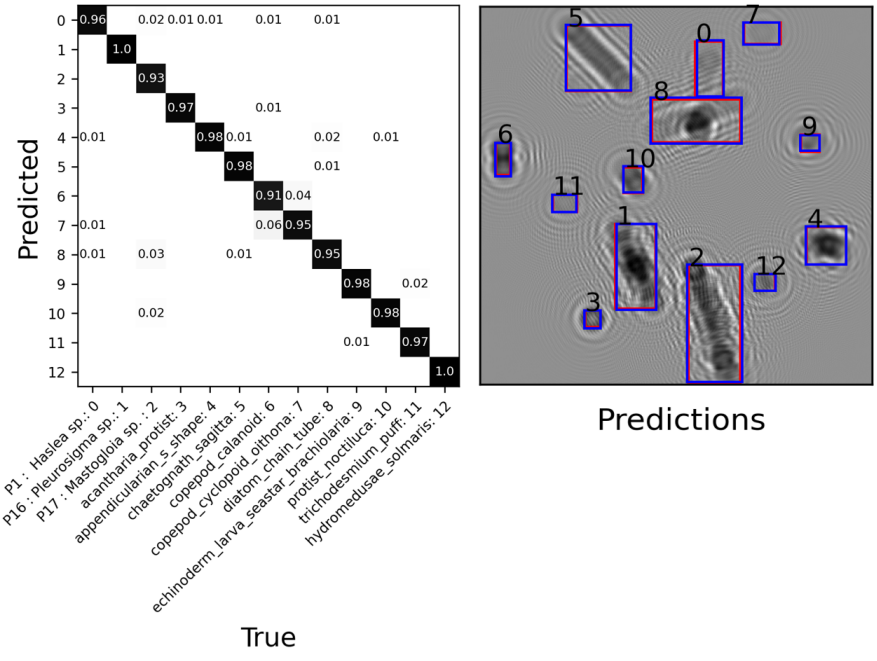

Fig. 7. Confusion matrix at IoU.5 and model predictions (blue : ground-truth , red : predicted).

3.2 Tracking performances

In this section we report the tracking performances performed by YOLO+SORT. We also computed the mean computation time of the whole pipeline when the real time detection (frame by frame) is performed by the smallest (v5s) and largest (v5x) versions of YOLOv5. The results of our evaluation are shown on Tab.2. The pipeline YOLOv5s+SORT can be used up to 44 FPS while the extra large version can be used up to 23 FPS. This difference is explained by the fact that the model has to make its inference at each frame and that for very large models it is often more optimized in term of speed to generate predictions on a batch of observations. The results suggest that the performances difference between the small and large version of YOLO is negligible when the input images are $t_0(x, y)$. When the input images are holograms, the use of a larger model improves the performances but the number of lost tracks remains higher than that of the models trained on transmission functions. However, the tracking performance on holograms remains high with for example a MOTA of 94.34% and 92.03% for YOLOv5x and YOLOv5s, respectively. An exemplary output of our pipeline is shown in Fig. 8. At each frame of the video, the total number of plankton per species can be updated. Note that we have slightly modified SORT, which is initially not class-aware, so that the predicted class of the object is saved as soon as a YOLO detection is associated with its tracker. To update the plankton count by class at a frame k, only plankton that were not detected in the past frames are added to the total count. When a plankton leaves the field-of-view of the video, the total count is not modified. For a plankton already detected in the previous frames, it is possible that YOLO predicts the wrong class during its trajectory. We therefore update the count by class by considering that the detected object has the class that obtained the maximum occurrence between frames 0 to $k - 1$.

Table 2. Tracking Performances.

Inputs	Model	MOTA	MOTP	MTR	MLR	FPS
Holograms	Yolov5s	92.03	84.76	92.54	1.94	44
	Yolov5x	94.35	86.33	95.30	1.43	23
$t_0(x, y)$	Yolov5s	96.16	88.89	96.32	0.72	-
	Yolov5x	96.05	90.66	96.63	0.92	-

4 Conclusion and perspectives

In this paper, we propose a pipeline that allows to detect, classify and count objects on raw holograms without going through the conventional holographic reconstruction/phase recovery steps. Our pipeline is composed of a real-time

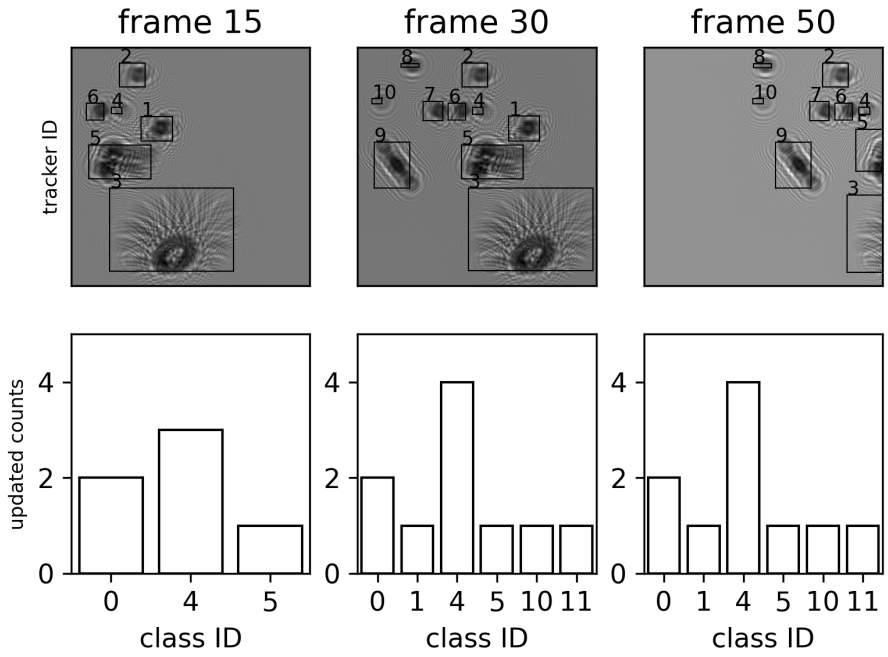

Fig. 8. Planckton tracked on a simulated video.

object detection model that performs the localization and classification of all objects present on the holograms and the SORT algorithm that links the detections through the video frames. We evaluated the object detection and tracking performances on simulated datasets that were generated with cropped plankton images obtained with a bright-field microscope and a shadow imager (ISIIS). Thirteen different species were considered for the simulations.

Two versions of YOLOv5 are trained to evaluate their detection performances on raw holograms. The results are compared with the detection performance obtained on transmission functions, which are the perfect images that the holographic reconstruction routine seeks to obtain. Note that in practice, obtaining a reconstructed holographic image of the same quality as our $t_0(x, y)$ in this paper is very complicated due to various noises and interferences on the hologram that can affect the conventional algorithms (focus/phase recovery) robustness. If anything, the presented comparison favors the holographic reconstruction/detection pipeline over the detection on raw hologram. However, although the results demonstrate that detection performances are slightly better on $t_0(x, y)$ than on holograms, the difference in AP@.5 is only 1.2 %. These results suggest that the prior realization of a holographic reconstruction, even perfectly conducted, does not significantly increase the performance of the object classification and detection tasks. With a AP@.5 score of 0.981, a YOLOv5x model can perform detection and classification of all plankton groups within a 512x512 raw hologram

(FOV ~ 0.33 mm^2) in a single pass in 14 ms. The tracking results show that the whole pipeline YOLOv5s+SORT can be performed in real-time (44 FPS) whereas YOLOv5x+SORT is slower (23 FPS) due to the large size of the model that required more floating-point operations suggesting that its usage could be more appropriate with batch (offline) tracking approaches.

Although the proposed method was validated with plankton images, it can be implemented to localize, count and identify other microscopic objects in raw holograms. Note that in practice, the object/camera distance was fixed at $z_2 = 0.8$ mm during our simulations. For three-dimensional imaging, the distance z_2 can vary from one plankton to another. This aspect is not addressed in this paper, which simply aims to show that holographic reconstruction is not necessary to detect, classify and track objects. With its current architecture, YOLOv5 is able to determine the (x,y,w,h) coordinates and the class of objects whose size may vary from a few pixels to a hundred pixels. To obtain the z-coordinate, the structure of the model could be modified. Otherwise, our pipeline is compatible with the recording of holograms. The bounding boxes provided by YOLOv5+SORT have the potential to facilitate the determination of the z-coordinate by any autofocusing algorithm.

While the results on simulated holograms are promising, it is often complicated and time consuming to put together a large dataset of real labeled holograms to train a detector. When a small labeled dataset is available, it might be beneficial to pre-train a detector with a large amount of simulated holograms and then use a transfer learning method to fine tune the model on the small dataset. Another approach would be to rely on an intensive data augmentation. Some works in the literature use de-focused back-propagated holograms as inputs of a deep learning model rather than raw holograms. By back-propagated the holograms on several planes near the correct global focus, the dataset could be significantly enlarged.

References

1. Bernardin, K., Stiefelhagen, R.: Evaluating multiple object tracking performance: The CLEAR MOT metrics. Eurasip Journal on Image and Video Processing **2008** (2008). https://doi.org/10.1155/2008/246309
2. Bewley, A., Ge, Z., Ott, L., Ramos, F., Upcroft, B.: Simple online and realtime tracking. Proceedings - International Conference on Image Processing, ICIP **2016-August**, 3464–3468 (2016). https://doi.org/10.1109/ICIP.2016.7533003
3. Bochkovskiy, A., Wang, C.Y., Liao, H.Y.M.: Yolov4: Optimal speed and accuracy of object detection. ArXiv **abs/2004.10934** (2020)
4. Lam, H.S., Tsang, P.W.: Invariant Classification of Holograms of Deformable Objects Based on Deep Learning. IEEE International Symposium on Industrial Electronics **2019-June**, 2392–2396 (2019). https://doi.org/10.1109/ISIE.2019.8781149
5. Latychevskaia, T.: Iterative phase retrieval in coherent diffractive imaging: practical issues. Applied Optics **57**(25), 7187 (2018). https://doi.org/10.1364/ao.57.007187
6. Latychevskaia, T., Fink, H.W.: Practical algorithms for simulation and reconstruction of digital in-line holograms. Applied Optics **54**(9), 2424 (2015)

7. Lee, S.J., Yoon, G.Y., Go, T.: Deep learning-based accurate and rapid tracking of 3D positional information of microparticles using digital holographic microscopy. Experiments in Fluids **60**(11) (2019). https://doi.org/10.1007/s00348-019-2818-y
8. Liu, X., Liu, X., Zhang, H., Fan, Y., Meng, H.: Research progress of digital holography in deep-sea in situ detection. Seventh Symposium on Novel Photoelectronic Detection Technology and Applications **11763**(March), 1760 —- 1766 (2021)
9. Memmolo, P., Iannone, M., Ventre, M., Netti, P.A., Finizio, A., Paturzo, M., Ferraro, P.: On the holographic 3d tracking of in vitro cells characterized by a highly-morphological change. Opt. Express **20**(27), 28485–28493 (Dec 2012). https://doi.org/10.1364/OE.20.028485
10. Memmolo, P., Miccio, L., Paturzo, M., Caprio, G.D., Coppola, G., Netti, P.A., Ferraro, P.: Recent advances in holographic 3D particle tracking. Advances in Optics and Photonics **7**(4), 713 (2015). https://doi.org/10.1364/aop.7.000713
11. Mudanyali, O., Tseng, D., Oh, C.: Compact, Light-weight and Cost-effective Microscope based on Lensless Incoherent Holography for Telemedicine Applications. Lab on Chip **10**(11), 1417–1428 (2010). https://doi.org/10.1039/c000453g.Compact
12. Padilla, R., Passos, W.L., Dias, T.L., Netto, S.L., Da Silva, E.A.: A comparative analysis of object detection metrics with a companion open-source toolkit. Electronics (Switzerland) **10**(3), 1–28 (2021)
13. Persson, J., Mölder, A., Sven-Göran Pettersson, P., Alm, K.: Cell motility studies using digital holographic microscopy. Microsc. Sci Technol. Appl. Edu **4** (01 2010)
14. Picart, P., Montresor, S.: Digital holography. Elsevier Inc. (2019). https://doi.org/10.1016/B978-0-12-815467-0.00005-0
15. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition **2016-Decem**, 779–788 (2016)
16. Rivenson, Y., Zhang, Y., Günaydın, H., Teng, D., Ozcan, A.: Phase recovery and holographic image reconstruction using deep learning in neural networks. Light: Science and Applications **7**(2), 17141 (2018)
17. Scholz, G., Mariana, S., Dharmawan, A.B., Syamsu, I., Hörmann, P., Reuse, C., Hartmann, J., Hiller, K., Prades, J.D., Wasisto, H.S., Waag, A.: Continuous live-cell culture imaging and single-cell tracking by computational lensfree LED microscopy. Sensors (Switzerland) **19**(5), 1–13 (2019). https://doi.org/10.3390/s19051234
18. Shimobaba, T., Kakue, T., Ito, T.: Convolutional Neural Network-Based Regression for Depth Prediction in Digital Holography. IEEE International Symposium on Industrial Electronics **2018-June**, 1323–1326 (2018)
19. Trujillo, C., Garcia-Sucerquia, J.: Automatic detection and counting of phase objects in raw holograms of digital holographic microscopy via deep learning. Optics and Lasers in Engineering **120**(August 2018), 13–20 (2019)
20. Wu, Y., Calis, A., Luo, Y., Chen, C., Lutton, M., Rivenson, Y., Lin, X., Koydemir, H.C., Zhang, Y., Wang, H., Göröcs, Z., Ozcan, A.: Label-Free Bioaerosol Sensing Using Mobile Microscopy and Deep Learning. ACS Photonics **5**(11), 4617–4627 (2018). https://doi.org/10.1021/acsphotonics.8b01109
21. Wu, Y., Rivenson, Y., Zhang, Y., Günaydin, H., Lin, X., Ozcan, A.: Extended depth - of - field in holographic image reconstruction using deep learning based auto - focusing and phase - recovery. Optica **5**, 704—-710 (2018)
22. Zhang, Y., Lu, Y., Wang, H., Chen, P., Liang, R.: Automatic classification of marine plankton with digital holography using convolutional neural network. Optics and Laser Technology **139**(January), 106979 (2021)