TrichTrack: Multi-Object Tracking of Small-Scale Trichogramma Wasps

Vishal Pani, Martin Bernet, Vincent Calcagno, Louise van Oudenhove, François F Bremond

TrichTrack: Multi-Object Tracking of Small-Scale *Trichogramma* Wasps

Vishal Pani[1,2], Martin Bernet[3], Vincent Calcagno[3,4], Louise van Oudenhove[3,4], and François Bremond[1,4]

[1]INRIA Sophia Antipolis - Méditerranée, France
[2]Indian Institute of Information Technology, Allahabad, India
[3]INRAE, CNRS, Institut Sophia Agrobiotech, France
[4]Université Côte d'Azur, France

Abstract

Trichogramma wasps behaviors are studied extensively due to their effectiveness as biological control agents across the globe. However, to our knowledge, the field of intra/inter-species Trichogramma behavior is yet to be explored thoroughly. To study these behaviors it is crucial to identify and track Trichogramma individuals over a long period in a lab setup. For this, we propose a robust tracking pipeline named TrichTrack. Due to the unavailability of labeled data, we train our detector using an iterative weakly supervised method. We also use a weakly supervised method to train a Re-Identification (ReID) network by leveraging noisy tracklet sampling. This enables us to distinguish Trichogramma individuals that are indistinguishable from human eyes. We also develop a two-staged tracking module that filters out the easy association to improve its efficiency. Our method outperforms existing insect trackers on most of the MOTMetrics, specifically on ID switches and fragmentations.

1. Introduction

Multi-Object Tracking (MOT) involves detecting and then tracking individual objects in a given video while having no prior information of the appearance or number of objects. The explosive progress in deep learning in the past decade has facilitated correlated progress in MOT [5] – enabling the application of MOT in a wide spectrum of fields. These can be tracking vehicles in the context of intelligent traffic systems [22, 10], pedestrians [7], animals for space-use studies [23] or intelligent livestock breeding [15], insects for behavioural studies [25, 1], etc.

In this paper, we focus on identifying different Trichogramma wasps that are indistinguishable to human eyes via ReID methods. We also identify and track Trichogramma wasps over a long period to facilitate the analysis of inter-species/sex behaviors. Our proposed method is also extendable to tracking any small-scale objects in a scene.

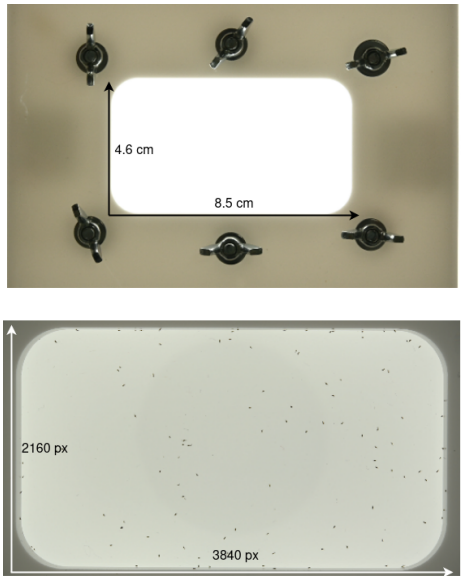

Figure 1: **Top**: Dimensions of the dish apparatus that contains the Trichogramma wasps during video capture. **Bottom**: Sample frame from a video containing over 100 individual wasps with the pixel dimensions.

To achieve this, it is natural for us to use existing state-of-the-art (SOTA) MOT trackers but we face four obstacles: (i) fully annotated Trichogramma videos are not available at present, preventing the use of supervised trackers [4, 2, 28]; (ii) existing works predominantly focus on tracking humans and commonplace objects, which makes them unsuitable for direct use in tracking Trichogramma without prior train-

ing; (iii) wasps in our dataset do not fly (as they are constrained to a dish) however they have an erratic motion with frequent, long, and random jumps, prohibiting the use of trackers which rely on the steady motion of the objects; (iv) wasps present in current videos are tiny when compared to the entire frame (each occupying about 0.03% of the total image area, see Fig 1 for dimensions), making their appearance features very sparse.

To overcome these obstacles, we adopt a weakly supervised training procedure for the detector on a small set of videos having noisy annotations provided by a fly tracking software called CTRAX [1]. We further fine-tune the TrichTrack detector on a larger set of videos with annotations from the earlier trained detector.

Since the wasps are very small relative to the entire frame, using end-to-end models [28, 18, 11] for tracking is sub-optimal. Hence, we use a robust two-staged online tracking module to associate the detections with identities. The two-staged tracker consists of easy associations followed by hard associations. The Easy Association Stage (EAS) improves the speed of the pipeline by filtering out detections that can be easily associated based on their Intersection over Union (IOU) with previous tracklets. The Hard Association Stage (HAS) associates the remaining detections by employing a motion and appearance model inspired by [27].

In the cases of jumps, the first stage, and the motion model of the second stage fail to associate the detections correctly. This happens because these associations are heavily dependent on the proximity of current detection and existing tracklets. However, the appearance of the wasp remains preserved while jumping. Hence, it is crucial to produce discriminative appearance vectors that allow the pipeline to track the wasps across jumps. For this, we train a ReID network based on a weakly supervised sampling technique, using noisy labels from CTRAX. This method expands upon the unsupervised sampling technique used in [14]. Since the training procedure uses triplet loss [12] to distinguish individuals, on inference, the model can produce discriminative feature vectors, helping the tracker to associate insects over jumps.

In summary, our contributions are as follows:

(i) We introduce a robust two-stage online tracker to handle the erratic movements of generic small-scale objects, in our case, Trichogramma wasps. Our tracker is also extendable to any scenario where objects may enter or exit the scene.

(ii) For training the ReID model, we use a weakly supervised sampling technique due to the unavailability of ground truth (GT) tracking annotations. Since the training on triplet loss, we can get distinguishable appearance vectors for the wasps.

(iii) By overcoming the problems posed by erratic movement and sparse features, our tracking pipeline outperforms SOTA insect and animal tracking softwares on MOT metrics [20, 7], specifically, MOTA, Identity (ID) switches, and fragmentations.

2. Related Work

2.1. Training ReID Networks

The unavailability of proper GT annotations necessitates the use of unsupervised or weakly supervised methods to train our ReID network. Fan et al. [9] propose an unsupervised progressive learning approach to transfer the pre-trained representation to new domains. But it is sensitive to the initial clusters produced from the pre-trained representations.

Meng et al. [19] formulate weakly supervised ReID as a multi-instance multi-label learning (MIML) problem. Further, Wang et al. [26] use a differentiable graphical model to tackle the problem of weakly supervised ReID. However, these formulations do not take advantage of noisy tracklets annotations in a MOT setting.

In this paper, we expand upon the unsupervised sampling technique used in [14]. Our contributions include the use of triplet loss with semi-hard and hard mining [12] to train the ReID model in a weakly supervised fashion. We also add a constraint that each sampled ID must exist in every frame of the sampling period. We discuss this in detail in Sec. 3.4.1.

2.2. Multi-Object Tracking

Multi-Object Offline Trackers [6, 29, 2, 21] give an optimal solution to the ID association problem since they have access to the entire video. However, the association problem being a minimum cost flow problem scales quadratically with the number of vertices. In our case, the number of Trichogramma individuals in one frame can exceed 120, making it infeasible to use such trackers. Instead, we look into Online Trackers which do not have access to future frames. These are usually faster than their offline counterparts. However, they may suffer in accuracy. Despite this, there has been significant progress in these trackers over the years. From using ReID networks to extract discriminative features [27, 24], to using transformers to track objects [18, 28, 3], to using differentiable renderers to predict the next frame [11], the literature in this topic is very rich. We use ReID networks since we have to extract features from small-scale objects that have sparse features.

2.3. Multi-Object Trackers for Insects

Over the years, trackers have been developed that focus solely on tracking insects. Of these, we find two of particular interest. **Ctrax [1]** uses the position and orientation of

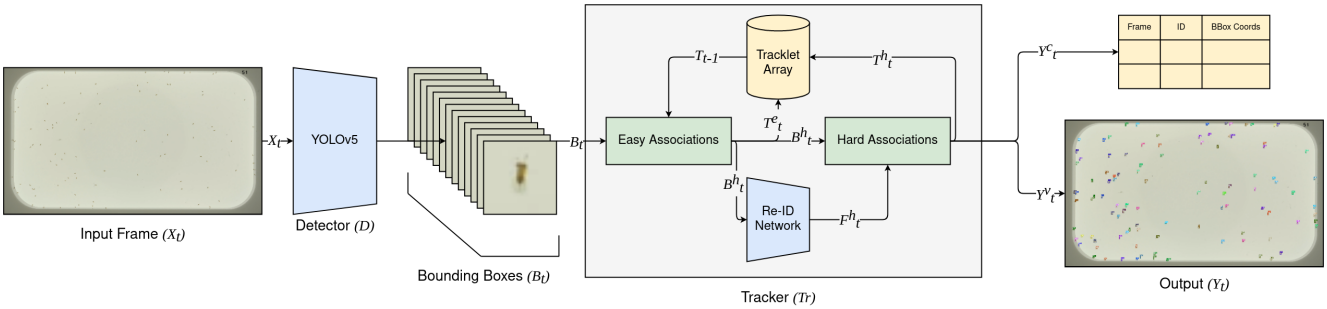

Figure 2: **Overview of the Pipeline, From Left to Right**: The input frame X_t is passed through the Detector D which generates a set of detected bounding boxes B_t. B_t is then fed into the tracker Tr which outputs Y_t containing the .csv output Y_t^c and video output Y_t^v.

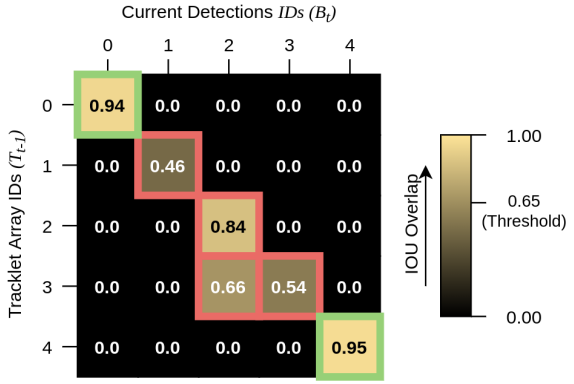

Figure 3: **IOU based association**: Only the cells with green borders are associated. Note, $b_{2,t}$ is not assigned to $tl_{2,t-1}$ because it overlaps with $tl_{3,t-1}$ too, therefore, violating the 2^{nd} condition.

insects to track them. However, it does not perform well in terms of accuracy and speed. **Trex [25]** uses a novel tree-based method for tracking along with the Hungarian algorithm. It is designed to focus on the speed of inference. But, there is scope to increase its accuracy.

3. Proposed Method

In this paper, we propose TrichTrack (TT). Since we follow the paradigm of tracking-by-detection, the pipeline comprises of two main modules: the detector and the tracker, see Fig. 2. The input to the detector is a video containing F number of frames represented by $X_v = \{X_1, X_2, ..., X_F\}$. The detector takes in the current RGB frame and outputs a set of detected bounding boxes. The tracker takes in the detected bounding box data and associates the current detections to the existing tracklet IDs. We provide more details of the modules in the next sections. Our primary contributions, which are the weakly supervised training procedure for the detector and the Re-ID network, are described in Sec. 3.3 - 3.4.

3.1. Detector

For a given time step t we have the input RGB frame X_t. X_t is passed through the Detector D (having learnable parameters θ_D) to get an array of detected bounding box coordinates B_t. The array of detected bounding boxes, $B_t = \{b_{1,t}, b_{2,t}, ..., b_{N,t}\}$, contains N individual bounding boxes, $b_{i,t}$, where $i \in \{1, 2, ..., N\}$. Each bounding box $b_{i,t}$ contains the x and y coordinates of the top-left and bottom-right vertices of the bounding box given by $b_{i,t} = [x^l_{i,t}, y^t_{i,t}, x^r_{i,t}, y^b_{i,t}]$. Where $x^l_{i,t}, x^r_{i,t}$ are the left and right x coordinates respectively, and $y^t_{i,t}, y^b_{i,t}$ are the top and bottom y coordinates respectively. It must be noted that the origin of the input X_t is the top-left corner.

3.2. Tracker

The tracker takes as input the detected array of bounding boxes B_t at given time step t and outputs a .csv file containing the positional details of the Trichogramma individuals and also a video displaying the tracked individuals along with their corresponding bounding boxes and identities. To understand the tracker we need to look deeper into its sub-modules, which are, Tracklet Array, EAS, ReID Network, and HAS; see Fig. 2 for reference.

3.2.1 Tracklet Array

The Tracker Array at a given time step t, $T_t = \{tl_{1,t}, tl_{2,t}, ..., tl_{M,t}\}$ holds updated states of M identified tracklets. A tracklet state of identity i at a given time step t is represented by, $tl_{i,t} = [\hat{x}^l_{i,t}, \hat{y}^t_{i,t}, \hat{x}^r_{i,t}, \hat{y}^b_{i,t}, \dot{\hat{x}}_{i,t}, \dot{\hat{y}}_{i,t}, \hat{f}_{i,t}]$. Where the first four values represent the top-left and bottom-right vertices of the bounding box, $\dot{\hat{x}}_{i,t}, \dot{\hat{y}}_{i,t}$ are the updated velocities in the x and y direction respectively, and

finally $\hat{f}_{i,t}$ is the updated feature vector of individual produced by the ReID network.

At time $t = 0$, for $tl_{i,1}$:

$$(\hat{x}^l_{i,1}, \hat{y}^t_{i,1}, \hat{x}^r_{i,1}, \hat{y}^b_{i,1}) = b_{i,1} \tag{1}$$

$$\dot{\hat{x}}_{i,1} = \dot{\hat{y}}_{i,1} = 0 \tag{2}$$

$$\hat{f}_{i,1} = RN(Pr_{ReID}(X_t[x^l_{i,1} : x^r_{i,1}, y^t_{i,1} : y^b_{i,1}]); \theta_{ReID}) \tag{3}$$

At time $t = 0$, the coordinates of the bounding box for tracklet i are set to the detected bounding box coordinates indexed at i. The velocities are set as 0. Also the feature vector is produced by the ReID network. Details regarding the ReID Network, RN, and the preprocessing stage, Pr, will be discussed in section 3.2.3.

Also, we follow an exponential moving average scheme to update the velocity and feature vector of the tracklet beyond $t = 0$. We set exponential coefficient as 0.5 in all of our experiments. Whenever $N > M$, we add the new detections to the tracklet array as mentioned in eqs. 1 - 3.

At any time step t we can get updated states of identity i by indexing into the tracklet array T_t at i to get $tl_{i,t}$.

3.2.2 Easy Association Stage

The EAS increases the efficiency of the tracker module and filtering out the easy matches between the detection of the current frame with the existing tracklets. We use intersection-over-union (IOU) based association to achieve this iff:

(i) The IOU of a detection, $b_{j,t}$, with a tracklet bounding box, $tl_{i,t}$, is more than a fixed threshold, γ.

(ii) The detection is not overlapping with any other tracklet bounding box, $tl_{k,t}$, where $k \in \{1, 2, ..., M\}$ and $k \neq i$.

Refer Fig. 3 for an illustrative representation of the process. Therefore, the EAS takes as input T_{t-1}, and B_t. It outputs the updated states of easily associated tracklets T^e_t and filters out the detected bounding boxes and tracklets for hard association, B^h_t and T^h_{t-1} respectively. This is given by:

3.2.3 ReID Network

The ReID Network (RN) is used in the HAS and is essential for handling jumps, as we can not use positional information for such cases. The details about the training of the ReID Network are mentioned in Section 3.4. Since the ReID handles square RGB crops, the rectangular crops of each detected bounding box in B^h_t must be transformed into square ones by padding with 0. Also, we apply image transformations such as sharpening and contrast correction (CLAHE) before inputting it to the RN. This preprocessing stage is represented by Pr_{ReID}. To increase efficiency, we actually input all of the preprocessed crops from B^h_t as a batch to the ReID model to get an array of feature vectors $F^h_t = \{f^h_{1,t}, f^h_{2,t}, ..., f^h_{N^h,t}\}$.

3.2.4 Hard Association Stage

In the HAS we use the Hungarian matching algorithm [16] to associate the remaining B^h_t with the remaining T^h_{t-1}. First we create a cost matrix $C_p \in [0, 1]^{M \times N}$ based on the euclidean distance between the positions of every $b^h_{i,t}$ with every $\hat{b}^h_{j,t-1}$. Then we create a cost matrix $C_a \in [0, 1]^{M \times N}$ based on the cosine distance between the features vectors of every $f^h_{i,t}$ with every $\hat{f}^h_{j,t-1}$ named. Therefore, a cell (m, n) in C_p is denoted as:

$$c^{m,n}_p = \frac{\sqrt{(x_{n,t} - \hat{x}_{m,t-1})^2 + (y_{n,t} - \hat{y}_{m,t-1})^2}}{\sqrt{2}} \tag{4}$$

Where,

$$x_{n,t} = \frac{x^l_{n,t} + x^r_{n,t}}{2}, \; \hat{x}_{m,t-1} = \frac{\hat{x}^l_{m,t-1} + \hat{x}^r_{m,t-1}}{2} \tag{5}$$

A cell (m, n) in C_a is denoted as:

$$c^{m,n}_a = \frac{||f^h_{n,t}||_2 \cdot ||\hat{f}^h_{m,t-1}||_2 - f^h_{n,t} \cdot \hat{f}^h_{m,t-1}}{2 \cdot ||f^h_{n,t}||_2 \cdot ||\hat{f}^h_{m,t-1}||_2} \tag{6}$$

We create the final cost matrix C_t, and use the Hungarian matching algorithm to associate detections indices, $I_N = \{i\hat{d}x_{1,t}, i\hat{d}x_{2,t}, ..., i\hat{d}x_{N_1,t}\}$, with tracklet indices, $I_M = \{idx_{1,t}, idx_{2,t}, ..., idx_{N_2,t}\}$. Where $N_1 = N_2 = M$, if $M < N$, else if $M > N$, $N_1 = N_2 = N$. These final steps are given by these equations:

$$C_t = \beta \cdot C_p + (1 - \beta) \cdot C_a \tag{7}$$

$$(I_m, I_n) = Hungarian(C_t) \tag{8}$$

For our experiments, $\beta = 0.7$ since jumps happen frequently but sporadically throughout the video. Finally, $idx_{1,t}$ is associated to $i\hat{d}x_{1,t}$, $idx_{2,t}$ to $i\hat{d}x_{2,t}$ and so on.

3.3. Training the Detector

Due to the availability of noisy GT as CTRAX annotations, we had to employ a weakly supervised method to train the detector. This process takes two iterations described below.

3.3.1 1st Iteration

Initially, the dataset contained only CTRAX annotations with lots of false negatives. Providing these false negatives

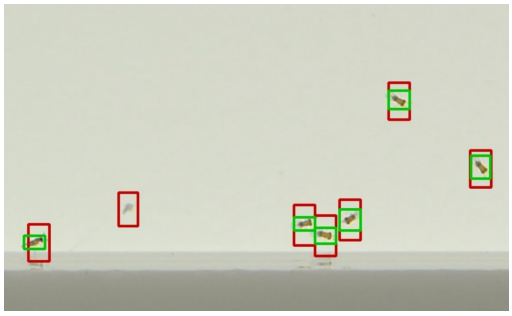

Figure 4: **Red Bounding Boxes**: Detection results with 1st iteration of training. **Green Bounding Boxes**: Detection results with 2nd iteration of training. The green bounding boxes are much tighter than red bounding boxes.

would have hampered the learning of the detector. So we masked them using OTSU masking algorithm [17]. This produces masked inputs which only contain true positives that help the detector to learn properly.

3.3.2 2nd Iteration

Since CTRAX annotations do not contain sides of the bounding boxes, the detector trained on the 1^{st} iteration could only produce loose bounding boxes. Loose bounding boxes are not desirable as they hamper the feature vectors produced by the ReID Network. To fix this, we trained the detector for a second time using annotations from the initially trained detector instead of CTRAX. We follow a data preprocessing scheme where we find out the contours of the insects and use them to filter only the center-most insect using OTSU algorithm to generate tight bounding box annotations. This helps the detector to produce tighter boxes. Refer Fig. 4 for results of the detector after 2^{nd} iteration of training.

3.4. Training the ReID Network

Our pipeline relies on the ReID Network for handling the cases of jumping and erratic movements. Therefore, it is essential to train the ReID Network properly. Due to the unavailability of GT tracking annotations, we use a weakly supervised method to train our ReID network via metric learning. To understand this more deeply we need to look closer into the sampling method for batches while training and the loss function. We discuss these in detail in the following subsections.

3.4.1 Sampling Method

First, we generate noisy tracklets from the CTRAX annotations. The sampling goal is to supply the ReID network with a proper batch during a training iteration. Since we use triplet loss for training, we must determine the positive and negative pairs too. Further, constraint the sampling to a continuous time period. We do so because jumps in the noisy annotations always result in ID switches. Hence, sampling over a discontinuous time period may result in incorrect pairing. Also, sampling in this manner simulates jumps when choosing positive pairs which are not consecutive to each other. Therefore, for frames at time period t to $t + \Delta$, we follow two criteria:

(i) Anchor $b_{i,t}$ forms a positive pair with $b_{j,t+\delta}$, iff, $i = j$ and $\exists\ b_{i,t+\delta}\ \forall\ \delta \in \{1, 2, ..., \Delta\}$.

(ii) Anchor $b_{i,t}$ forms a negative pair with $b_{j,t+\delta}$, iff, $i \neq j$ and $\exists\ b_{j,t+\delta}\ \forall\ \delta \in \{1, 2, ..., \Delta\}$.

Now to decide the size of the batch we choose the number of different Trichogramma per batch, C, where $C < M$ and number of instances to be randomly sampled from each Trichogramma individual, S, where $S < \Delta$. Finally, batch size bs, is the product of C and S. The entire sampling process is visualized in Fig. 5.

3.4.2 Loss Function

After determining a proper batch to feed into the training step of the ReID network, we first train the network with online semi-hard for some epochs to get a competent ReID nentwork. And then finetune it by training again with online hard triplet mining. Online hard triplet loss is given as follows:

$$\mathcal{L}_{\text{BH}}(bs; \theta_{RN}) = \overbrace{\sum_{i=1}^{P}\sum_{a=1}^{K}}^{\text{all anchors}} [m - \overbrace{\min_{\substack{j=1...P \\ n=1...i \\ j\neq i}} D\left(f^i_{a;\theta_{RN}}, f^j_{n;\theta_{RN}}\right)}^{\text{hardest negative}} + \overbrace{\max_{p=1...K} D\left(f^i_{a;\theta_{RN}}, f^i_{p;\theta_{RN}}\right)}^{\text{hardest positive}}]_+ \tag{9}$$

Where, m is the margin defined during training, we use $m = 1$ in all of our experiments. For online semi-hard mining we first find out the pairs which satisfy the following conditions and use the above equation without find the minimum negative pair and maximum positive pair:

$$||f^i_{a;\theta_{RN}} - f^i_{p;\theta_{RN}}||_2 + m < ||f^i_{a;\theta_{RN}} - f^j_{n;\theta_{RN}}||_2 \tag{10}$$

Here, f^i_a is of the anchor insect with identity i, f^i_p is the positive pair to f^i_a and f^j_n is the negative pair with identity j.

We use a pre-trained ResNet-18 for the ReID Network. We remove the last classification layer and add a fully connected layer with 128 nodes. Therefore, for every patch we get a feature vector, $f_{\theta_{RN}} \in \mathbb{R}^{128\times 1}$.

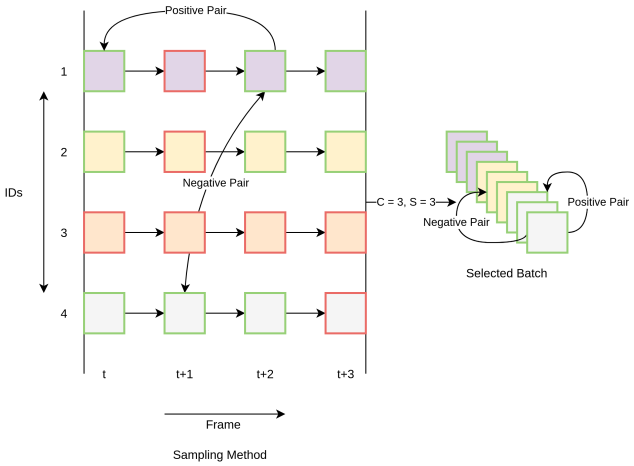

Figure 5: **Sampling Process for ReID Network, Left to Right**: Here we have C = 3, S = 3, Δ = 4. The detections with green borders are sampled, whereas the ones with red borders are rejected. Examples of positive and negative pairs are also shown. The selected batch contains 9 trichogramma crops.

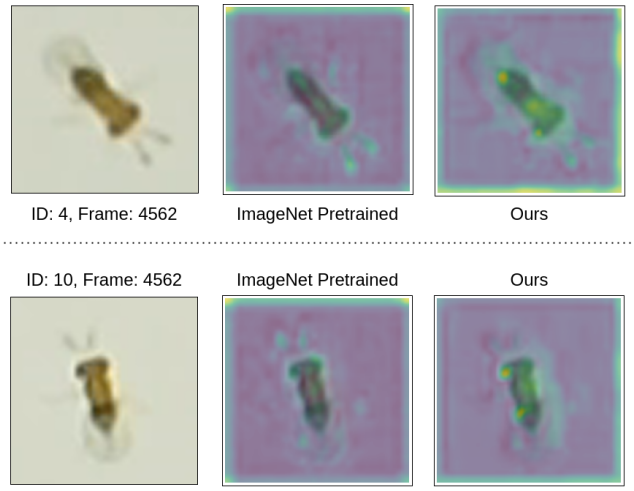

Figure 6: **Comparison of Saliency Maps. Middle:** Generated by ImageNet pre-trained network. **Right:** Generated by our ReID network.

4. Dataset

The data was collected by culturing the strains of Trichogramma and recording them in a dish-like apparatus by biologists, as shown in the top of Fig 1. We work with two species: Cacoeciae, and Brassicae. Each species contain three strains. The recorded videos and CTRAX annotations are from density experiments. High-density experiments contain more than 100 individuals in a given frame, whereas low-density experiments contain as few as 12 individuals per frame. Also, each video is about 8 minutes long and has 25 fps. Therefore, for each video, we have about 12,000 frames. We use 1 GT video of each strain, having an average of 15 individuals per frame, produced by manual annotation. They are named (*format: strain_density_num*), ISA3080_Low_13, E1.3_Low_7, ACJY_Low_14, PMbio1_Low_11, PJ_Low_8, and PR002_Low_12.

5. Results and Analysis

Before the results and analysis, we first discuss the implementation details and the algorithms used to draw comparisons in the results and analysis.

5.1. Implementation Details

For this work, we aim to run the pipeline on a previous generation gaming computer. Therefore, we restrict the training and inference of the pipeline to a GTX 1080ti graphics card: with 11 GBs for VRAM, and an Intel Xeon Processor E5-2630 v3 CPU node: providing a processor frequency of 2.40 GHz. In this paper, we use YOLOv5 as our detector as it has one of the highest accuracy to speed ratios amongst state-of-the-art detectors [13].

5.2. Algorithms

Throughout the Experiments sections, we will draw comparisons between the following algorithms:

(i) **CTRAX**: This is an insect tracking algorithm that uses the position and orientation of insects to track.

(ii) **TREX**: This is an animal tracking algorithm that uses a tree-based method for tracking along with the Hungarian algorithm.

(iii) **Baseline**: The baseline algorithm uses YOLOv5 as the detector and DeepSORT as the tracker to track the Trichogramma individuals.

(iv) **TrichTrack**: The proposed algorithm uses YOLOv5 and a modified tracker to track the Trichogramma individuals.

5.3. Experiments

5.3.1 Saliency Maps by ReID Network

To plot the saliency maps generated by the ReID Networks, we take the channel-wise average of the output feature map of the sixth intermediate block of ResNet-18. We use PMbio1_Low_11 for this experiment.

On referring to Fig. 6, we observe that the activations of the saliency maps produced by our ReID Network are

Video: PMbio1_Low_11										
Algo.	**GT**	**MT**	**PT**	**ML**	**FP**	**FN**	**IDs**	**FM**	**MOTA**	**MOTP**
CTR	14	5	5	4	0	72010	169	220	53.10%	40.90%
TRX	14	7	6	1	0	37956	60	1820	75.30%	12.80%
Bln	14	4	9	1	15	36801	1741	3309	75.00%	53.60%
TTr	14	**14**	**0**	**0**	**0**	**43**	**0**	**7**	**99.80%**	**99.90%**
Video: ISA3080_Low_13										
Algo.	**GT**	**MT**	**PT**	**ML**	**FP**	**FN**	**IDs**	**FM**	**MOTA**	**MOTP**
CTR	18	12	5	1	**65**	29039	576	634	84.50%	31.20%
TRX	18	17	1	0	5307	4999	1814	1789	93.70%	12.20%
Bln	18	16	1	1	1494	12268	3787	3442	90.80%	19.99%
TTr	18	**18**	**0**	**0**	1742	**744**	**324**	**408**	**98.50%**	**55.40%**
Video: PR002_Low_12										
Algo.	**GT**	**MT**	**PT**	**ML**	**FP**	**FN**	**IDs**	**FM**	**MOTA**	**MOTP**
CTR	12	6	6	0	51	37382	220	290	73.80%	33.50%
TRX	12	12	0	0	5145	2063	338	794	94.80%	12.70%
Bln	12	7	5	0	26	25643	909	2410	81.50%	37.70%
TTr	12	**12**	**0**	**0**	**0**	**260**	**34**	**64**	**99.80%**	**78.00%**

[a]CTR = CTRAX, Bln = Baseline, TRX = TREX, TTr = TrichTrack.
[b]$MT \uparrow$, $PT \downarrow$, $ML \downarrow$, $FP \downarrow$, $FN \downarrow$, $IDs \downarrow$, $FM \downarrow$, $MOTA \uparrow$, $MOTP \uparrow$.

Table 1: Evaluation of algorithms on MOTMetrics. TrichTrack is consistently the best performing tracker.

more pronounced around the antenna, eye, thorax, and abdomen of the Trichogramma individual when compared to the results by the ImageNet pre-trained network. Thereby, verifying that our model attends to salient features of the individual. This facilitates the production of discriminative feature vectors of the individual. This result also indicates a new avenue of research for insect ReID. Since now we know the salient features of the Trichogramma individuals, future works can exploit it to develop better ReID networks for insects with sparse features.

5.3.2 Detection Performance

To analyze the effectiveness of our detector sub-module, we compare its detection mean Average Precision (mAP) as per Pascal VOC metric [8] on the three videos with CTRAX and Trex. Based on Table 2, we see that TrichTrack outperforms previous methods by a large margin. The other methods produce low scores because of high False Negatives (FN) as can be seen in Table 1.

5.3.3 Performance on MOT Metrics

Table 1 contains the MOT metrics evaluations of 3 videos and all 4 algorithms. The most crucial metrics for our study are: (i) Mostly Tracked (MT), (ii) False Positives (FP), (iii) False Negatives (FN), (iv) ID Switches (IDs), (v) Fragmentations (FM), (vi) MOT Accuracy (MOTA), and (vii) MOT Precision (MOTP). We see that TrichTrack performs the best consistently in most of the metrics.

$\text{mAP}^{\text{IOU=0.5}}$			
Videos	**CTRAX**	**Trex**	**TrichTrack**
ISA3080_Low_13	11.22	56.81	**97.15**
PMbio1_Low_11	12.19	55.79	**98.97**
PR002_Low_12	9.06	69.22	**98.81**

Table 2: Evaluation of detection performance for each algorithm.

We see that TrichTrack performs extremely well in eliminating IDs and FM. Also, it has significantly higher MOTA and MOTP than the other methods and much lower FN. This shows the detector detects up most of the Trichogramma individuals in a given frame. Refer 1 for a more detailed look into the metric evaluations.

6. Conclusion

In this work, we have developed a robust two-stage online tracker to handle the erratic movements of generic small-scale objects, in our case, Trichogramma wasps. Due to the unavailability of labeled data, we trained our detector using an iterative weakly-supervised method that preprocessed noisy data before training the network. Based on our training procedure, the detector successfully reduced

False Negatives. We also used a weakly supervised method to train our ReID network that leverages conditional noisy tracklet sampling to drive its training process. Due to this, our ReID model attends to salient features of the feature-sparse wasps and produces highly discriminative feature vectors. Also, we developed a two-staged tracking module that filters out the easy associations to improve the efficiency of the tracker. Quantitatively, our tracker performed better on detection mAP and MOTMetrics when compared with existing trackers dedicated to insect tracking. Specifically, we improved upon the ID switches, false positives, fragmentations, and MOT precision.

References

[1] K. Branson, A. A. Robie, J. Bender, P. Perona, and M. H. Dickinson. High-throughput ethomics in large groups of drosophila. *Nature Methods*, 6(6):451–457, Jun 2009. 1, 2

[2] G. Braso and L. Leal-Taixe. Learning a neural solver for multiple object tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 1, 2

[3] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko. End-to-end object detection with transformers, 2020. 2

[4] P. Chu, J. Wang, Q. You, H. Ling, and Z. Liu. Transmot: Spatial-temporal graph transformer for multiple object tracking, 2021. 1

[5] G. Ciaparrone, F. L. Sánchez, S. Tabik, L. Troiano, R. Tagliaferri, and F. Herrera. Deep learning in video multi-object tracking: A survey. *CoRR*, abs/1907.12740, 2019. 1

[6] A. Dehghan, S. M. Assari, and M. Shah. GMMCP-Tracker:globally optimal generalized maximum multi clique problem for multiple object tracking. In *CVPR*, 2015. 2

[7] P. Dendorfer, S. H. Rezatofighi, A. Milan, J. Shi, D. Cremers, I. D. Reid, S. Roth, K. Schindler, and L. Leal-Taixé. CVPR19 tracking and detection challenge: How crowded can it get? *CoRR*, abs/1906.04567, 2019. 1, 2

[8] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, June 2010. 7

[9] H. Fan, L. Zheng, C. Yan, and Y. Yang. Unsupervised person re-identification: Clustering and fine-tuning. *ACM Trans. Multimedia Comput. Commun. Appl.*, 14(4), Oct. 2018. 2

[10] Z. He, Y. Lei, S. Bai, and W. Wu. Multi-camera vehicle tracking with powerful visual features and spatial-temporal cue. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 203–212, 2019. 1

[11] Z. He, J. Li, D. Liu, H. He, and D. Barber. Tracking by animation: Unsupervised learning of multi-object attentive trackers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 2

[12] A. Hermans, L. Beyer, and B. Leibe. In defense of the triplet loss for person re-identification. *CoRR*, abs/1703.07737, 2017. 2

[13] G. Jocher, A. Stoken, J. Borovec, NanoCode012, ChristopherSTAN, L. Changyu, ..., and P. Rai. ultralytics/yolov5: v3.1 - Bug Fixes and Performance Improvements, Oct. 2020. 6

[14] S. Karthik, A. Prabhu, and V. Gandhi. Simple unsupervised multi-object tracking, 2020. 2

[15] W. Kim, Y. B. Cho, and S. Lee. Thermal sensor-based multiple object tracking for intelligent livestock breeding. *IEEE Access*, 5:27453–27463, 2017. 1

[16] H. W. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97, 1955. 4

[17] D. Liu and J. Yu. Otsu method and k-means. In *2009 Ninth International Conference on Hybrid Intelligent Systems*, volume 1, pages 344–349, 2009. 5

[18] Z. Lu, V. Rathod, R. Votel, and J. Huang. Retinatrack: Online single stage joint detection and tracking. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 2

[19] J. Meng, S. Wu, and W. Zheng. Weakly supervised person re-identification. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 760–769, 2019. 2

[20] A. Milan, L. Leal-Taixé, I. Reid, S. Roth, and K. Schindler. MOT16: A benchmark for multi-object tracking. *arXiv:1603.00831 [cs]*, Mar. 2016. arXiv: 1603.00831. 2

[21] I. Papakis, A. Sarkar, and A. Karpatne. Gcnnmatch: Graph convolutional neural networks for multi-object tracking via sinkhorn normalization. *CoRR*, abs/2010.00067, 2020. 2

[22] Y. Qian, L. Yu, W. Liu, and A. G. Hauptmann. Electricity: An efficient multi-camera vehicle tracking system for intelligent city. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2020. 1

[23] A. Rathore, A. Sharma, N. Sharma, and V. Guttal. Multi-object tracking in heterogeneous environments (mothe) for animal space-use studies. *bioRxiv*, 2020. 1

[24] B. Shuai, A. Berneshawi, X. Li, D. Modolo, and J. Tighe. Siammot: Siamese multi-object tracking. In *CVPR*, 2021. 2

[25] T. Walter and I. D. Couzin. Trex, a fast multi-animal tracking system with markerless identification, and 2d estimation of posture and visual fields. *eLife*, 10:e64000, feb 2021. 1, 3

[26] G. Wang, G. Wang, X. Zhang, J. Lai, Z. Yu, and L. Lin. Weakly supervised person re-id: Differentiable graphical learning and a new benchmark. *IEEE Transactions on Neural Networks and Learning Systems*, 32:2142–2156, 2021. 2

[27] N. Wojke, A. Bewley, and D. Paulus. Simple online and realtime tracking with a deep association metric. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 3645–3649. IEEE, 2017. 2

[28] F. Zeng, B. Dong, T. Wang, C. Chen, X. Zhang, and Y. Wei. Motr: End-to-end multiple-object tracking with transformer, 2021. 1, 2

[29] J. Zhou, G. Cui, Z. Zhang, C. Yang, Z. Liu, and M. Sun. Graph neural networks: A review of methods and applications. *CoRR*, abs/1812.08434, 2018. 2