

Article

Efficient Real-Time Droplet Tracking in Crop-Spraying Systems

Truong Nhut Huynh ¹, Travis Burgers ^{2,3} and Kim-Doang Nguyen ^{1,*}¹ Florida Institute of Technology, Melbourne, FL 32901, USA; nhuynh2023@my.fit.edu² CNH Industrial, Sioux Falls, SD 57104, USA; travis.burgers@cnh.com³ South Dakota State University, Brookings, SD 57007, USA

* Correspondence: knguyen@fit.edu

Abstract: Spray systems in agriculture serve essential roles in the precision application of pesticides, fertilizers, and water, contributing to effective pest control, nutrient management, and irrigation. These systems enhance efficiency, reduce labor, and promote environmentally friendly practices by minimizing chemical waste and runoff. The efficacy of a spray is largely determined by the characteristics of its droplets, including their size and velocity. These parameters are not only pivotal in assessing spray retention, i.e., how much of the spray adheres to crops versus becoming environmental runoff, but also in understanding spray drift dynamics. This study introduces a real-time deep learning-based approach for droplet detection and tracking which significantly improves the accuracy and efficiency of measuring these droplet properties. Our methodology leverages advanced AI techniques to overcome the limitations of previous tracking frameworks, employing three novel deep learning-based tracking methods. These methods are adept at handling challenges such as droplet occlusion and varying velocities, ensuring precise tracking in real-time potentially on mobile platforms. The use of a high-speed camera operating at 2000 frames per second coupled with innovative automatic annotation tools enables the creation of a large and accurately labeled droplet dataset for training and evaluation. The core of our framework lies in the ability to track droplets across frames, associating them temporally despite changes in appearance or occlusions. We utilize metrics including Multiple Object Tracking Accuracy (MOTA) and Multiple Object Tracking Precision (MOTP) to quantify the tracking algorithm's performance. Our approach is set to pave the way for innovations in agricultural spraying systems, offering a more efficient, accurate, and environmentally responsible method of applying sprays and representing a significant step toward sustainable agricultural practices.

Keywords: droplet tracking; crop-spraying systems; deep learning; real-time detection; high-speed camera; automatic annotation; data augmentation



Citation: Huynh, T.N.; Burgers, T.; Nguyen, K.-D. Efficient Real-Time Droplet Tracking in Crop-Spraying Systems. *Agriculture* **2024**, *14*, 1735. <https://doi.org/10.3390/agriculture14101735>

Academic Editor: Francesco Marinello

Received: 12 July 2024

Revised: 24 September 2024

Accepted: 28 September 2024

Published: 2 October 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In modern agriculture, spray nozzles are essential for accurately delivering treatments such as pesticides and fertilizers to crops. The efficiency and effectiveness of these systems largely depend on understanding the characteristics of the droplets they release, particularly their size and velocity [1]. These properties are critical for maximizing treatment efficacy and minimizing environmental impacts such as chemical runoff. However, existing methods for measuring droplet properties, including patternators and immersion sampling, often suffer from limited accuracy, high costs, and complex implementation processes [2,3].

To address these challenges, we introduce a novel technique that accurately measures droplet size and enables real-time tracking and velocity calculation using advanced video analysis. Our approach leverages deep learning to strive for a significant advancement in accessibility, cost-effectiveness, and precision, with the potential to revolutionize nozzle design and optimize treatment efficiency in agriculture.

Deep learning algorithms, particularly those designed for object detection and tracking, can significantly enhance the accuracy of droplet measurement. By considering factors

such as droplet distance, velocity, and appearance, these algorithms can provide precise tracking even under challenging conditions such as varying light and movement. Our research introduces a benchmark dataset for droplet tracking created using high-speed video footage of droplets in various conditions. This dataset is crucial for training and evaluating our system and for ensuring that it performs accurately in real-world agricultural scenarios. Central to our study is the application of deep learning, a potent tool capable of real-time multi-droplet detection and tracking. By integrating a deep learning-based algorithm that considers droplet distance, velocity, and appearance, we are able to significantly enhance tracking accuracy under various conditions. The results demonstrate our method's remarkable accuracy and efficiency, emphasizing its role in promoting sustainable agricultural practices.

While the primary test scenario involved fog droplets, chosen for their ability to simulate high particle density and dynamic movement patterns, the detection and tracking algorithms developed in this study are designed to be generalizable. These algorithms are adaptable to a range of droplet types, including those found in agricultural spraying systems. This adaptability is critical for ensuring the feasibility of applying the developed methods to real-world crop spraying scenarios.

Before detailing our proposed methodology, we review the existing literature on droplet measurement and tracking in agricultural systems. This review highlights the limitations of current methods and sets the stage for our innovative solution, which addresses these shortcomings by enhancing precision, accessibility, and cost-effectiveness. We discuss various challenges in accurate droplet measurement and real-time tracking, including the complexities of droplet dynamics and the need for robust detection methods.

2. Literature Review

Particle Image Velocimetry (PIV) is currently the leading technique for measuring fluid flows, including droplet dynamics. PIV works by visualizing and calculating fluid velocities [4]. In a PIV system, tiny particles are introduced into the fluid and illuminated by a laser sheet. A high-speed camera captures images of these illuminated particles, then PIV software (PIVlab, version 2.56, and DaVis, version 10.1) analyzes successive frames using cross-correlation techniques to determine particle displacements. These displacement data are converted into velocity vectors, providing a spatially resolved map of the fluid velocity field. Despite PIV's ability to offer valuable and non-intrusive data on fluid velocities, its application in measuring and tracking individual droplets remains limited due to challenges in resolving small-scale droplet behaviors. Moreover, PIV requires a controlled lab setting, as the use of laser illumination necessitates close proximity to the equipment, making it impractical for outdoor or large-scale agricultural environments. Despite significant advancements, traditional computer vision techniques also fall short in addressing the complexities of droplet behavior in agricultural sprays [5–7]. These methods often struggle with issues such as droplet occlusion, varying droplet velocities, and the need for high precision in dynamic environments. This gap in the current technology highlights the need for a more sophisticated approach that integrates recent advances in deep learning with computer vision.

The application of deep learning in object detection and tracking has seen significant strides in recent years, becoming pivotal in areas such as autonomous vehicles and robotics. Techniques such as Faster R-CNN [8], YOLO [9], SSD [10], RetinaNet [11], and Mask R-CNN [12] have set new benchmarks in object detection and tracking. These methods have not only proven their efficacy in various applications but have also opened new avenues for research and development in deep learning.

Recent studies in this field have been geared towards enhancing the accuracy and efficiency of object detection and tracking in challenging environments [13–19]. Acharya et al. (2022) [20] laid the groundwork for applying deep learning to droplet detection and tracking in agricultural spraying. However, their approach has several limitations:

- The method employed by Acharya et al. struggles to maintain high accuracy without sacrificing processing speed, making it less suitable for real-time applications in dynamic environments.
- Their study primarily focused on detection accuracy, with limited attention to comprehensive tracking metrics such as Multiple Object Tracking Accuracy (MOTA) and Multiple Object Tracking Precision (MOTP) that are critical for evaluating real-world performance.
- Acharya et al. relied on manually annotated datasets, which is time consuming and prone to human error, limiting the scalability and consistency of their approach.
- Their paper did not thoroughly address the overall inference time of the pipeline, which is a critical factor for real-time applications.

To address these limitations, the current paper introduces significant advancements in droplet tracking:

- Design of improved deep learning architectures to precisely track droplets moving across the camera frame.
- Enhanced droplet tracking accuracy with reduced processing time, surpassing the balance achieved in [20].
- Introduction of a comprehensive set of metrics for evaluating tracking performance.
- Extraction of the dynamic properties of droplets, focusing on their velocity.
- Development of an autonomous annotation method for large-scale droplet data, addressing the impracticality of manual labeling.
- Optimization of the pipeline's overall inference time.

Statement of Contributions

In this work, we harness the capabilities of deep learning and computer vision to effectively detect and track the motion of droplets sprayed from agricultural nozzles in real time. Our algorithm leverages the latest advancements in detection and tracking methods, such as YOLOv8 [21], OcSORT [22], StrongSORT [23], and BotSORT [24], to facilitate real-time, precise, and reliable performance. We have successfully addressed notable challenges prevalent in prior frameworks, including droplet occlusion and variations in droplet size and shape.

Key to our approach is the implementation through deep learning models of a robust algorithm that is adept at capturing complex visual patterns even in the presence of occlusions. This enables precise detection and tracking of droplets irrespective of partial obstructions or concealment. Additionally, we develop a flexible framework capable of adapting to variations in droplet size and shape by dynamically adjusting detection parameters based on observed droplet characteristics, ensuring consistent accuracy across diverse conditions.

Our experimental setup included an array of crop spray nozzles and a high-speed camera system for capturing detailed droplet motion. The data annotation process, which is a significant challenge due to the sheer volume of droplets, was streamlined using an automatic annotation method with YOLOv8, resulting in an extensive and accurately labeled droplet dataset.

The contributions of our work are as follows:

- Development and deployment of a highly efficient droplet tracking system capable of real-time performance.
- Development of an AI-driven tool specifically designed to handle droplet trajectories to automate the annotation and augmentation of the tracking data.
- Substantial optimization of state-of-the-art tracking models, ensuring faster and more accurate droplet tracking.
- Novel approaches to overcome the limitations of existing AI-based frameworks in multi-droplet tracking tasks.

- Introduction of a novel method to extract the geometric and dynamic properties of droplets, including their size, area, and velocity.
- Deployment of the proposed framework on high-performance computing systems, including a supercomputer, as well as an onboard platform (Jetson Orin). This dual deployment enhances scalability and enables edge AI capabilities for real-time crop spraying system analysis in mobile/field tasks as well as large-scale simulations in controlled environments.

3. Real-Time Deep Learning-Based Object Tracking

Most existing related work focuses on developing algorithms for detecting droplets in an image, which is essential for measuring droplet size. Tracking of droplets, on the other hand, is crucial for extracting droplet velocity information from a droplet video. This problem was addressed in [20] by employing the Kalman filter and correspondence matching. However, this method has certain limitations. First, it is unable to detect occluded droplets, resulting in inability to identify droplets before and after occlusion with the same identification. Consequently, a single droplet before and after occlusion is considered as two separate droplets. Furthermore, the Kalman filter assumes a constant velocity of the droplet, resulting in imprecise tracking.

To overcome these limitations, we design and implement three deep learning-based tracking techniques to further enhance the precision and accuracy of droplet tracking. These methods enable our framework to efficiently operate in real time on a mobile platform. The general tracking procedure is illustrated in Figure 1, and proceeds as follows:

- **Detecting Droplets in the Initial Video Frame:** The tracking process begins with the identification of droplets in the first frame. This step involves analyzing the image or video frame to detect regions corresponding to droplets. Various object detection techniques, including deep learning-based approaches and feature-based methods, can be employed to accurately detect and localize droplets within the frame (see Figure 1, first panel).
- **Assigning Identities to Detected Droplets:** To establish identity continuity, each detected droplet is assigned a unique identity or label. This step ensures that the same droplet can be correctly associated in subsequent frames despite changes in appearance or occlusions. Identity assignment is achieved by associating a unique identifier, such as a numerical value or unique tag, with each detected droplet (see Figure 1, second and third panels).
- **Tracking and Temporal Data Association:** The core of the droplet tracking framework lies in following the detected droplets across frames and associating them temporally. By employing techniques such as Kalman filtering, particle filtering, and data association algorithms, the proposed framework establishes correspondences between droplets in the current frame and those in the subsequent frames. This temporal data association allows for tracking of individual droplets over time, enabling analysis of their motions, interactions, and behaviors (see Figure 1, fourth panel).
- **Real-Time Operation on Mobile Platform:** The developed deep learning-based tracking technique enables precise and accurate droplet tracking, allowing the framework to operate efficiently in real time on a mobile platform.

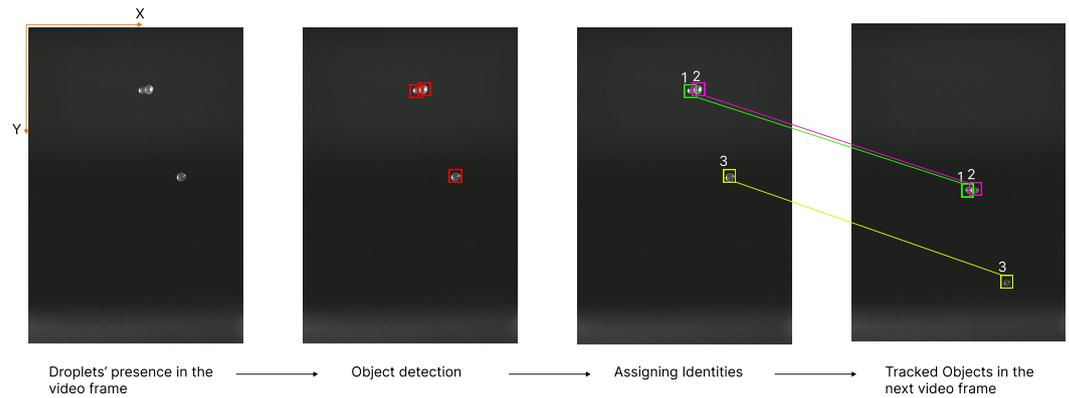


Figure 1. Multiple object tracking procedure.

The rest of this section elaborates on the three methods grounded in state-of-the-art computer vision tracking that we use for tracking the droplets.

3.1. Method 1: BoT-SORT Extension for Droplet Tracking (BSET-DT)

This subsection introduces our BoT-SORT Extension for Droplet Tracking (BSET-DT), which adapts the BoT-SORT methodology described in [24] with significant modifications and refinements to address the unique challenges and requirements associated with droplet tracking. Our BSET-DT integrates motion and appearance information, camera motion compensation, and an enhanced Kalman filter state vector, all of which are tailored to the nuances of droplet dynamics. The key steps in the method are presented below.

3.1.1. Step 1: Advanced Droplet Detection and Re-Identification

The foundational phase of our BSET-DT methodology involves a sophisticated iterative detection process engineered using a state-of-the-art detection algorithm \mathcal{D} . This algorithm is pivotal in analyzing each frame \mathbf{F}_t of the video sequence to yield bounding boxes \mathbf{B}_t , class probabilities \mathbf{C}_t , and confidence scores \mathbf{S}_t . The detection equation is iteratively applied across the video sequence, ensuring comprehensive analysis and detection accuracy:

$$(\mathbf{B}_t, \mathbf{C}_t, \mathbf{S}_t) = \mathcal{D}(\mathbf{F}_t), \quad \forall t \in \{1, 2, \dots, T\}. \quad (1)$$

The re-identification module \mathcal{R} further enhances this process by encoding distinct appearance features of each detected droplet in frame \mathbf{F}_t . This encoding is critical in differentiating and re-identifying droplets over successive frames, especially in densely populated droplet environments. The feature extraction mechanism is robust, ensuring the uniqueness of each droplet's representation. The features used for re-identification include:

- **Shape and Size:** The geometric properties of droplets, such as their bounding box dimensions and aspect ratio.
- **Color and Texture:** Visual characteristics captured through histograms and texture descriptors.
- **Motion Patterns:** Temporal changes in the position and velocity of droplets across frames.

For example, the feature vector for a droplet might include the dimensions of its bounding box, a histogram of pixel intensities within the bounding box, and a vector of optical flow magnitudes indicating its movement between frames.

The encoding process can be represented as follows:

$$\mathbf{F}_t = \mathcal{R}(\mathbf{B}_t, \mathbf{F}_t), \quad \forall t \in \{1, 2, \dots, T\}. \quad (2)$$

This advanced detection and re-identification strategy ensures that each droplet is not only detected with high accuracy but also consistently tracked across frames. The robustness of this methodology lies in its ability to maintain the integrity of droplet identities

even in scenarios with complex droplet dynamics and interactions. Furthermore, the use of advanced machine learning models in \mathcal{D} and \mathcal{R} leverages deep learning advancements for enhanced detection and feature encoding.

3.1.2. Step 2: Advanced Fusion of Motion and Appearance Data

In this pivotal step, our BSET-DT framework innovatively integrates motion analysis with appearance data to forge a comprehensive tracking strategy. This integration is crucial in addressing complex tracking scenarios where both motion dynamics and visual features play significant roles.

Kalman Filter for Motion Prediction

We employ the discrete Kalman filter \mathcal{K} with a constant velocity model, a widely used approach for object motion modeling in image planes. This model is particularly effective in predicting the trajectory of droplets in a controlled manner. The state vector \mathbf{X}_t and covariance matrix \mathbf{P}_t are iteratively updated for each frame t , capturing the motion dynamics of droplets. The update equations are provided as follows:

$$\mathbf{X}_{t+1} = \mathbf{A}\mathbf{X}_t + \mathbf{B}\mathbf{U}_t + \mathbf{W}_t \quad (3)$$

$$\mathbf{P}_{t+1} = \mathbf{A}\mathbf{P}_t\mathbf{A}^T + \mathbf{Q}, \quad \forall t \in \{1, 2, \dots, T\} \quad (4)$$

where \mathbf{A} and \mathbf{B} represent the state transition and control matrices, \mathbf{U}_t is the control vector, and \mathbf{W}_t and \mathbf{Q} denote the process noise and its covariance matrix. This formulation is derived from improvements in recent tracking methods in which direct estimation of the bounding box dimensions has shown better performance.

Integrating Appearance Features

Simultaneously, we leverage the appearance features \mathbf{F}_t extracted from the re-identification module. These features encapsulate crucial visual information about each droplet, enabling the system to maintain consistent identification across frames. The fusion of motion and appearance data is mathematically represented as follows:

$$\mathbf{T}_t = \text{Fuse}(\mathbf{X}_t, \mathbf{F}_t), \quad \forall t \in \{1, 2, \dots, T\} \quad (5)$$

where \mathbf{T}_t symbolizes the unified tracking information for each droplet in frame t , combining motion predictions from the Kalman filter with the visual features from the appearance model. This fusion enhances tracking accuracy, particularly in scenarios with complex motion patterns and varying visual appearances.

The advanced fusion methodology in our BSET-DT ensures robust tracking performance even in challenging conditions, and effectively handles the intricacies of droplet dynamics and appearances.

3.1.3. Step 3: Advanced Camera-Motion Compensation

Recognizing the critical role of camera movement in the accuracy of droplet tracking, especially in potential mobile applications and on-machine recordings over bumpy terrain, our BSET-DT integrates a sophisticated camera motion compensation mechanism \mathcal{M} . This advanced module is instrumental in adjusting the detected positions of droplets to counteract the effects of camera motion, thereby preserving the fidelity of the tracking process in dynamic scenarios.

Addressing Dynamic Camera Movements

In dynamic camera scenarios such as those encountered in MOT20 [25], the location of the bounding boxes in the image plane can shift dramatically due to camera movement, potentially leading to increased identity switches or false negatives. Even in static camera settings, motion caused by external factors such as wind can have a substantial impact.

Our approach to Camera Motion Compensation(CMC) accounts for both rigid motion associated with changes in camera pose and non-rigid motion of objects within the scene.

Global Motion Compensation Technique

To effectively compensate for camera motion, we employ the Global Motion Compensation (GMC) technique, which involves the following steps:

1. **Keypoint Extraction:** Initially, key points within the image are extracted to serve as critical markers for tracking motion across frames.
2. **Sparse Optical Flow:** The extracted key points are then tracked using sparse optical flow, enabling the determination of motion vectors that encapsulate the camera's movement.
3. **Affine Transformation:** Based on the motion vectors, an affine transformation matrix is computed and used to adjust the bounding boxes. This transformation corrects for the camera's rigid motion, effectively stabilizing the image with respect to background motion.

The compensation equation for each frame is provided by

$$\mathbf{B}_{\text{adjusted},t} = \mathcal{M}(\mathbf{B}_t, \mathcal{C}_{\text{camera},t}), \quad \forall t \in \{1, 2, \dots, T\}, \quad (6)$$

where \mathbf{B}_t denotes the detected bounding boxes in frame t , $\mathcal{C}_{\text{camera},t}$ represents the affine transformation matrix encapsulating the camera motion characteristics at frame t , and $\mathbf{B}_{\text{adjusted},t}$ indicates the compensated bounding boxes for that frame.

Enhancing Tracking Accuracy

By meticulously compensating for camera movement, this advanced CMC approach ensures that tracking accuracy is not compromised by extraneous camera motions, thereby enhancing the reliability and effectiveness of our BSET-DT in diverse and dynamic tracking environments, especially in the field. Algorithm 1 summarizes the tracking procedure.

Algorithm 1 Pseudocode for BoT-SORT Extension for Droplet Tracking (BSET-DT)

```

1: Input: Video sequence  $\{\mathbf{F}_t\}_{t=1}^T$ 
2: Initialize an empty set of tracks
3: for  $t = 1$  to  $T$  do
4:   // Step 1: Advanced Droplet Detection and Re-Identification
5:    $(\mathbf{B}_t, \mathbf{C}_t, \mathbf{S}_t) = \mathcal{D}(\mathbf{F}_t)$ 
6:    $\mathbf{F}_t = \mathcal{R}(\mathbf{B}_t, \mathbf{F}_t)$ 
7:   // Step 2: Advanced Fusion of Motion and Appearance Data
8:   Update  $\mathbf{X}_t$  and  $\mathbf{P}_t$  using Kalman filter  $\mathcal{K}$ 
9:    $\mathbf{T}_t = \text{Fuse}(\mathbf{X}_t, \mathbf{F}_t)$ 
10:  // Step 3: Advanced Camera-Motion Compensation
11:  Extract keypoints and compute sparse optical flow
12:  Compute affine transformation  $\mathcal{C}_{\text{camera},t}$ 
13:   $\mathbf{B}_{\text{adjusted},t} = \mathcal{M}(\mathbf{B}_t, \mathcal{C}_{\text{camera},t})$ 
14:  // Track Management and Update
15:  for each detected droplet do
16:    Associate detections to existing tracks or initialize new tracks
17:    Update track states with adjusted bounding boxes
18:  end for
19:  // Post-Processing for Enhanced Tracking
20:  Apply track verification and filtering techniques
21:  Refine trajectories and generate visualizations
22: end for
23: Output: Final set of droplet tracks

```

We adapt BoT-SORT's robust tracking framework to handle the unique characteristics of droplets, such as their small size and rapid unpredictable movements, which are different from pedestrian tracking scenarios. The re-identification model in BoT-SORT was fine-tuned for droplet tracking, ensuring that droplets can be accurately re-identified across frames even when they exhibit significant shape and size changes. BoT-SORT's motion model was modified to predict the movement of droplets more accurately, taking into account factors such as wind influence and spray patterns which are not considered in pedestrian tracking.

3.2. Method 2: Observation-Centric Droplet Tracking (OCDT)

This subsection introduces our Observation-Centric Droplet Tracking (OCDT) methodology, which is inspired by the OC-SORT approach [22] but has been significantly modified and refined to address the unique challenges and requirements associated with droplet tracking. The OCDT methodology is based on the idea that the accuracy of detections is crucial when it comes to tracking. Focusing on precise detections of high quality is a strategic shift from conventional tracking methods, which tend to prioritize factors such as motion prediction or data association. The OCDT method aims to accomplish high-quality observations by leveraging the droplet detection algorithm proposed in our previous work [26], which was designed to reduce the inference time while maintaining the detection accuracy in order to enhance tracking accuracy and robustness. The key steps in this method are discussed below.

3.2.1. Step 1: Droplet Detection

Our droplet tracking algorithm starts by employing the droplet detector developed in [26] to detect droplets on each input video frame. The droplet detector outputs the predicted bounding boxes (B), class probabilities (C), and confidence scores (S) for each detected droplet as follows:

$$(B, C, S) = \text{Droplet Detector}(\text{Frame}). \quad (7)$$

3.2.2. Step 2: Droplet Feature Extraction

After the droplets are detected, we extract discriminative features from the detected droplets. These features encapsulate the essential characteristics of the droplets, including their size, shape, and texture. To achieve this, we utilize a type of deep convolutional neural network called a Re-Identification (ReID) [27] model that has been trained to encode these attributes into high-dimensional feature representations. The ReID model is particularly well suited for this task due to its ability to learn and extract robust and discriminative features from droplets even under varying conditions. This ensures that the extracted features are not only representative of the droplets' characteristics but are also robust to changes in lighting, perspective, camera setup, and other environmental factors.

These high-dimensional feature representations play a vital role in the subsequent stages of data association and tracking. They provide a basis for comparing droplets across different frames, enabling the identification of the same droplet in consecutive frames. The feature extraction process can be represented as follows:

$$F = \text{ReID}(\text{Frame}, B) \quad (8)$$

where F represents the features extracted from the droplets detected in the frame. Through this process, we are able to transform the raw droplet detections into a rich set of features, setting the stage for the subsequent steps in our methodology.

3.2.3. Step 3: Droplet Data Association

In our tracking method, the data association step is a crucial component that utilizes the extracted features to establish robust associations between droplets across consecutive frames. We employ a two-pronged approach that combines appearance and motion cues

to ensure accurate and reliable temporal data associations for droplet tracking. For the appearance, we compute the cosine similarity to measure the similarity between two vectors F_i and F_j (i.e., feature vectors coming from (8)) that represent droplets in consecutive frames i and j , providing the cosine of the angle between the two droplet vectors, which represents their orientation. This metric measures the matching cost between the feature set of the droplet and the feature of the detection during the association process.

On the other hand, for the motion cues we employ the Kalman filter, a powerful tool for predicting the expected locations of droplets in the current frame based on their previous positions and the physics of motion. The Kalman filter works by iteratively updating the estimates of droplet positions as new video frames become available. This effectively minimizes the estimation error by incorporating both the new measurements and the motion predictions in a statistically optimal way, as described in Algorithm 2. We also measure the spatiotemporal dissimilarity between the motion state of droplets and incoming new detections using the Mahalanobis distance. This distance metric takes into account the covariance of the data, and can effectively handle the inherent motion and occlusion challenges present in droplet tracking.

The cost matrix for droplet association is then computed as a weighted sum of the appearance cost and the motion cost. This fusion of appearance and motion cues provides a comprehensive measure for data association, ensuring that the matched pairs of droplets between frames are visually similar and have consistent motion patterns. The data association process in our method can be represented as follows:

$$(Matches, Predictions) = DataAssociation(F, PreviousTracks) \quad (9)$$

where *Matches* indicates the matched pairs of droplets between frames and *Predictions* denotes the predicted locations of droplets in the current frame.

To maintain the integrity and continuity of the droplet tracks, we incorporate a track management technique. This technique predicts the positions of existing droplet tracks based on estimated motion models and applies track fragmentation and merging strategies to handle occlusions or interruptions, helping to ensure the reliable and seamless tracking of droplets. Furthermore, we integrate track verification and filtering techniques to enhance the reliability and accuracy of tracked droplets. These techniques include track consistency checks such as motion coherence and spatial consistency as well as confidence estimation methods. By validating the quality of droplet tracks and filtering out false positives or noisy tracks, our approach significantly improves the accuracy and robustness of the tracked droplets.

Track verification can be achieved by assessing the consistency of the predicted tracks with respect to their motion and spatial attributes. Specifically, the predicted velocity of a droplet track can be compared to the average velocity of nearby droplet tracks using the following equation:

$$Consistency(t) = |Velocity(t) - AvgVelocity(t - 1 : t + 1)| \quad (10)$$

where $Velocity(t)$ represents the predicted velocity of the droplet track at time t and $AvgVelocity(t - 1 : t + 1)$ denotes the average velocity of nearby droplet tracks at times $t - 1$ to $t + 1$. If the consistency value exceeds a predefined threshold, the droplet track can be considered inconsistent and subjected to further filtering.

Confidence estimation is another important aspect of track verification. By assigning a confidence score to each droplet track, the algorithm can differentiate between reliable and uncertain droplet tracks. Confidence is estimated based on the track length, spatial accuracy, and feature similarity with other droplet tracks. The confidence score is calculated as follows:

$$Confidence(t) = TrackLength(t) \times SpatialAccuracy(t) \times FeatureSimilarity(t) \quad (11)$$

where $\text{TrackLength}(t)$ represents the length of the droplet track at time t , $\text{SpatialAccuracy}(t)$ indicates the spatial accuracy of the droplet track, and $\text{FeatureSimilarity}(t)$ reflects the similarity between the droplet track's features and the features of nearby tracks [23]. By combining consistency checks and confidence estimation, the tracker effectively verifies and filters the droplet tracks, ensuring that only reliable and meaningful droplet tracks are retained for further analysis.

3.2.4. Step 4: Track Maintenance and Termination

Track maintenance and termination are vital components in our OCDT methodology, helping to ensure that the tracking process remains efficient, accurate, and adaptable to changing conditions. This step involves the following key aspects:

Track Maintenance: For tracks that do not receive a matching detection in the current frame, our method applies motion models or prediction algorithms to update the track's state estimation and retains their ID in case there are matches in future frames. The method includes mechanisms to manage the age and visibility of tracks, including incrementing the age of a track for each frame and updating visibility metrics based on recent detections. Additionally, our OCDT method dynamically adjusts tracking parameters (e.g., similarity thresholds and motion models) based on observed conditions.

Track Termination: If a track does not receive any matching detection for a defined number of consecutive frames, then it is marked as inactive or deleted from the set of tracks. If a track moves outside a defined region of interest or violates spatial constraints, then it is terminated. Our OCDT employs specific criteria for track termination, including low confidence scores, erratic motion patterns, or inconsistent appearance features. Our algorithm concludes with postprocessing steps to refine the tracked droplet trajectories and generate insightful visualizations. Trajectory analysis techniques, including interpolation and smoothing, are employed to enhance the visual quality and interpretability of the droplet tracking results. Additionally, the droplet tracker enables seamless integration with other applications or systems, facilitating further analysis and interactions with the tracked droplets. Algorithm 2 explains the OCDT tracking procedure in detail.

OCDT leverages a droplet detection algorithm from our previous work optimized for the detection of small and fast-moving objects. This is an enhancement over the general object detection used in OC-SORT, which may not perform as well in specialized tasks such as droplet tracking. OCDT introduces advanced feature extraction using a Re-Identification (ReID) model specifically tuned for droplets. This model captures fine-grained features such as shape, size, and texture that are critical for accurately re-identifying droplets across frames, something that is not emphasized in OC-SORT. OCDT improves upon OC-SORT's data association by combining appearance-based cues with motion-based predictions. The integration of a Kalman filter for motion prediction and use of the Mahalanobis distance for spatiotemporal matching enhance the algorithm's ability to maintain accurate droplet identities across frames even in challenging conditions. OCDT introduces more sophisticated track management techniques tailored to the unique challenges of droplet tracking, including consistency checks and confidence estimation.

Algorithm 2 Pseudocode explaining droplet tracking Method 2: OCDD Tracker

Input: Video frames

- 1: Initialize an empty set of tracks.
- 2: **for** each frame in the video sequence **do**
- 3: Perform object detection to obtain a set of bounding box detections for the current frame.
- 4: **for** each detection **do**
- 5: Find the best matching track for the detection based on chosen similarity metrics (IOU and appearance similarity).
- 6: **for** each track t in the set of tracks **do**
- 7: Compute the similarity score $sim(t, \text{detection})$ based on the chosen metric.
- 8: **end for**
- 9: Assign the detection to the track with the highest similarity score, if it exceeds a predefined threshold.
- 10: **if** a matching track is found and satisfies certain criteria **then**
- 11: Update the matched track with the current detection.
- 12: Update the track's state estimation using the Kalman filter equations:

$$\begin{aligned} \text{Prediction: } x(t) &= F \cdot x(t-1) + w(t-1), \\ P(t) &= F \cdot P(t-1) \cdot F^T + Q(t-1), \\ \text{Update: } K(t) &= P(t) \cdot H^T \cdot (H \cdot P(t) \cdot H^T + R)^{-1}, \\ x(t) &= x(t) + K(t) \cdot (z(t) - H \cdot x(t)), \\ P(t) &= (I - K(t) \cdot H) \cdot P(t), \end{aligned}$$
- 13: Update any other relevant track attributes (e.g., age, visibility).
- 14: **end if**
- 15: **if** no matching track is found or the matching track does not meet the criteria **then**
- 16: Initialize a new track with the current detection.
- 17: Assign a unique ID to the new track.
- 18: Initialize the track's state estimation using the detection information.
- 19: Add the new track to the set of tracks.
- 20: **end if**
- 21: **end for**
- 22: Perform track maintenance.
- 23: **for** each unassigned track **do**
- 24: Update the track's state estimation using motion models or prediction algorithms.
- 25: **end for**
- 26: **for** each track **do**
- 27: **if** the track does not receive any matching detection for a defined number of frames **then**
- 28: Mark the track as inactive or delete it from the set of tracks.
- 29: **end if**
- 30: **if** the track goes outside **then**
- 31: the defined region of interest
- 32: Terminate the track and remove it from the set of tracks.
- 33: **end if**
- 34: **end for**
- 35: **end for**
- 36: Output the final set of tracks along with their associated states and any additional information.

Output: Tracked droplet trajectories

3.3. Method 3: Droplet Tracker via Adaptation of StrongSORT

In this subsection, we describe our Droplet Tracker via Adaptation of StrongSORT (DTAS) method for precise and reliable droplet tracking in video sequences. This approach, while inspired by the foundation laid by the StrongSORT algorithm [23], has been adapted to better suit our needs and tailored to the unique challenges of droplet tracking. We walk through the key steps in our method, including preprocessing, object detection, feature extraction, data association, track management, track verification

and filtering, and postprocessing. Each of these stages has been carefully designed and implemented to substantially improve the overall accuracy and reliability of our droplet tracking system. The following description provides a detailed walkthrough of each step in Algorithm 3, highlighting the key components and techniques used for precise and reliable droplet tracking.

- **Preprocessing:** The video frames are preprocessed by resizing, denoising, and normalizing to prepare them for object detection.
- **Initialization:** An empty list of tracks is initialized to store the droplet trajectories.
- **Droplet Detection:** For each frame in the video, droplets are detected using a customized droplet detector trained specifically for this purpose.
- **Feature Extraction:** Features are extracted from the detected objects using the Omni-Scale Feature Re-Identification (ReID) model, providing robust and discriminative representations of the droplets.
- **Track Creation and Management:**
 - **First Frame:** If the frame is the first frame, then a new track is created for each detected object.
 - **Subsequent Frames:** Data association is performed using a two-pronged approach that combines appearance and motion cues. The association metric is calculated based on the similarity between existing tracks and detected objects.
 - **Track Update:** Existing tracks are updated based on the associations, and unmatched detections are evaluated to create new tracks.
- **Position Prediction:** The position of existing tracks is predicted using a state transition model.
- **Handling Occlusions and Interruptions:** Techniques such as track fragmentation and merging strategies are applied to handle occlusions or interruptions in the tracking process.
- **Track Verification and Filtering:** The tracks are verified and filtered to remove potential false positives and noisy tracks. This step involves consistency checks and confidence estimation methods to ensure the quality of the tracked droplets.
- **Postprocessing and Visualization:** Postprocessing and visualization techniques are applied to enhance the presentation of the tracked droplet trajectories.
- **Output:** The final output of the algorithm consists of the tracked droplet trajectories, providing a comprehensive and accurate representation of the droplets' movements throughout the video sequence.

DTAS includes specific preprocessing steps such as denoising and normalization that are critical for handling the high variability in droplet appearance due to environmental factors. DTAS uses a customized droplet detector trained specifically for identifying droplets in video frames. This detector is more specialized than the general object detectors used in StrongSORT, which may not perform as well on small and fast-moving objects such as droplets. Similar to OCDT, DTAS employs an Omni-Scale Feature Re-Identification (ReID) model to extract robust features from detected droplets. This is coupled with an advanced data association strategy that merges appearance and motion cues, ensuring accurate tracking even under occlusion or rapid movement conditions. DTAS introduces advanced track management strategies, including handling occlusions and interruptions, which is critical for maintaining continuous tracking in dynamic environments. These features extend the capabilities of StrongSORT, making it more suitable for droplet tracking.

Algorithm 3 Pseudocode explaining droplet tracking Method 3

Input: Video frames

- 1: Preprocess video frames (resize, denoise, normalize)
- 2: Initialize an empty list of tracks
- 3: **for** each frame in video frames **do**
- 4: Detect objects in the current frame
- 5: Extract features from the detected objects
- 6: **if** it's the first frame **then**
- 7: **for** each detection in Detected objects **do**
- 8: Create a new track with the detection
- 9: **end for**
- 10: **else**
- 11: Perform data association

$$\text{Association metric} = \max_{i,j} (\text{similarity}(t_i, d_j))$$

- 12: Update the existing tracks based on the associations
- 13: **for** each unmatched detection in Unmatched detections **do**
- 14: **if** meetsCertainCriteria(unmatched detection) **then**
- 15: Create a new track with the unmatched detection
- 16: **end if**
- 17: **end for**
- 18: **end if**
- 19: Predict the position of existing tracks

$$\hat{x}_{k|k-1} = F \cdot x_{k-1|k-1}$$

- 20: Handle occlusions or track interruptions
- 21: Update the tracks with the new detections
- 22: Verify and filter the tracks to remove potential false positives or noisy tracks
- 23: Perform any desired post-processing or visualization
- 24: **end for**
- 25: function meetsCertainCriteria(detection):
- 26: **if** (detection.confidence > confidenceThreshold and detection.size within sizeRange and motionConsistent(detection) and withinSpatialConstraints(detection)) **then**
- 27: return true
- 28: **else:**
- 29: return false
- 30: **end if**

Output: Tracked droplet trajectories

4. Annotation and Augmentation of Experimental Data

The datasets used and analyzed during the current study are available from the corresponding author upon reasonable request. For the successful development of a deep learning-based model capable of precise and robust droplet tracking, the creation of a new dataset tailored to meet the following specific criteria is imperative:

- **Varied Droplet Dynamics:** The dataset must encompass a wide array of droplet characteristics, including differences in size, shape, and movement patterns. This diversity is essential for training a robust detection algorithm to accurately generalize across various droplet behaviors and adapt to unpredictability.
- **High-Quality Imaging Standards:** The collected images should span a range of lighting conditions and backgrounds, and should be captured using an assortment of imaging techniques. Inclusion of images taken from multiple perspectives, distances, and contextual settings enhances the robustness of the detection algorithm under real-world conditions.

- **Multiplicity of Droplet Scenes:** It is crucial that the dataset include instances where numerous droplets are present within a single frame. Such complexity challenges the model to effectively track multiple targets and discriminate between adjacent droplet paths annotated with distinct bounding boxes and identifiers.
- **Ample Training Volume:** The dataset should contain a substantial volume of annotated images for the training and validation phases. Data of sufficient breadth are vital for allowing the model to develop reliable feature recognition capabilities, in particular to accommodate the extensive parameter space typical of deep learning models.

4.1. Self-Supervised Annotation Process

To fulfill the outlined prerequisites, we embarked on an extensive data collection initiative, capturing an extensive collection of images and videos of droplets released from a variety of nozzles. A comprehensive database was then constructed from these visuals, delineated into subsets for training, validation, and testing. Assembling such an extensive trove of droplet data is an arduous endeavor, made particularly laborious due to the manual annotation required for each droplet across thousands of frames. As this manual process is characterized by its sluggish pace, susceptibility to human error, and limitations in scalability, it presents a bottleneck.

To streamline this labor-intensive process, we propose an integrated solution: an automatic annotation tool conceived to operate in tandem with the deep learning model established earlier in our study. Initially, we laid the groundwork by manually annotating a modest subset of droplet data which served to train an initial droplet detection model. This precursor model was then deployed to autonomously identify and categorize droplets within new images, methodically assigning bounding boxes and labels. The subsequent phase involved manually refining these preliminary annotations to ensure accuracy. While manual intervention is not entirely eliminated, its role is significantly diminished, with the bulk of the annotation workload shouldered by the automatic tool.

The implementation of this tool for droplet tracking marks a paradigm shift in annotation efficiency, drastically curtailing the time and labor traditionally required to curate datasets of considerable magnitude. Such enhanced productivity is indispensable for managing the vast quantities of droplet instances encountered in our research. At this advanced stage, the detector's proficiency negates the need for manual labeling in training droplet tracking models, with the exception of the necessary task of associating individual droplet identities.

4.2. Data Augmentation

Additionally, image augmentation techniques were implemented to augment the dataset, thereby increasing its size and diversity. Prior studies have demonstrated that image augmentation yields improved classification and detection outcomes, as evidenced by the controlled experiments conducted in [19,28]. In this work, we utilized three specific data augmentation methods: (1) Cutout, which involves removing random sections of the image, effectively "cutting out" patches to enhance generalization; (2) Noise, where random noise is added to the image to help the model become more robust to slight variations; and (3) Mosaic, a technique that combines parts of different images into a single image, creating a mosaic pattern to help the model recognize objects in various contexts. The effects of these augmentations are illustrated in Figure 2. Cutout augmentation works by randomly masking out sections of the image during training. This forces the model to learn to detect and classify objects even when part of the object or background is missing, leading to better generalization. As shown in the figure, the model trained with Cutout augmentation tends to focus on detecting the most prominent droplets. However, compared to the Mosaic and Noise augmentations, the model might miss smaller or partially obscured objects, as it has been trained to expect that portions of the image can be occluded. While this technique can help to reduce overfitting, there is a tradeoff in sensitivity, as the model might ignore smaller droplets that are not fully visible or ones that appear in regions frequently masked during

training. Despite this, Cutout augmentation can enhance the robustness of the model by improving its ability to make predictions from incomplete data, which can be particularly useful in real-world scenarios where objects might be occluded or only partially visible. By introducing the Noise augmentation, the model learns to become more resilient to small variations in the data that do not change the fundamental features of the objects. In the context of droplet detection, this augmentation helps the model focus on the key features that define a droplet, rather than being misled by minor image distortions. In the figure, the magenta boxes indicate the detected droplets; while the model performs well, it appears that introduction of the Noise augmentation may have led to a slight reduction in the number of detections compared to the Mosaic-augmented model. Mosaic augmentation is the most effective technique for enhancing dataset diversity. Mosaic augmentation involves stitching together parts of different images to create a new composite image, with objects appearing in various positions, scales, and contexts. This augmentation technique allows the model to see objects in different configurations, which leads to improved generalization when detecting objects in varying conditions. In Figure 2a, the model trained with Mosaic augmentation produces the highest number of detections, as indicated by the numerous green boxes around the droplets. The model is able to detect droplets of various sizes and in different regions of the image, suggesting that Mosaic augmentation helps the model to become more versatile. By combining parts of different images, Mosaic augmentation encourages the model to learn features that are invariant to changes in position, scale, and context, leading to better overall performance in detecting a wide variety of objects. As a result, the Mosaic augmentation technique was identified as the optimal method for enhancing classification and tracking performance in this project. The parameters for the Cutout and Noise techniques were determined from the existing literature and our initial experimental evaluations. Specifically, the Cutout technique involved removing 10% of random sections from the images, while the Noise technique added 5% random noise to the images. These settings were selected to avoid excessive data distortion that could hinder the model's learning process. While it is conceivable that more aggressive augmentation (e.g., greater than 10% removal in Cutout or addition of more than 5% Noise) could lead to different outcomes, in this study we found the Mosaic augmentation technique to be the most effective.

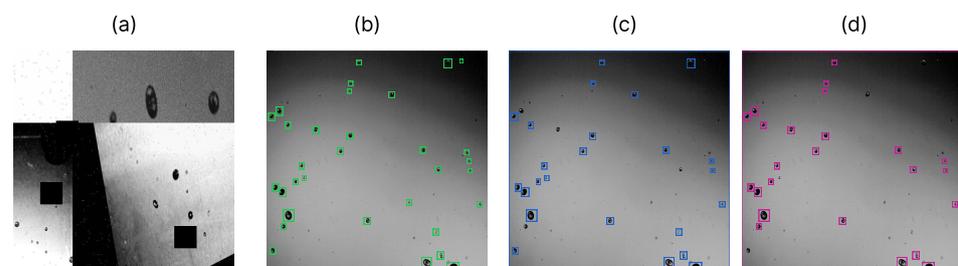


Figure 2. Effects of different data augmentation techniques on model predictions. (a) Illustration of the Mosaic augmentation technique used to enhance dataset diversity. Model predictions after training with different augmentation methods: (b) Mosaic augmentation, (c) Cutout augmentation, and (d) Noise augmentation.

4.3. Data Organization

The database was divided into three distinct subsets, namely, the training set, validation set, and testing set. The training set is composed of images employed for training the AI models, enabling them to learn and optimize their parameters. The validation set is dedicated to evaluating the performance and fine-tuning of the models during the training process. Finally, evaluation of the model's performance on the validation set, particularly through the assessment of the loss metric, plays a pivotal role in gauging its generalization capabilities. The loss metric provides valuable insights into how effectively the model can apply the acquired knowledge to previously unseen droplet data. Consequently, the performance analysis on the validation set serves as valuable guidance for

making informed decisions pertaining to parameter updates and regularization techniques aimed at enhancing the model's overall performance and effectiveness. The testing set consists of images that are entirely unseen by the model, serving to provide an independent assessment of its generalization capabilities. Our newly constructed dataset represents a substantial expansion compared to the dataset utilized in our previous work [20], amounting to twelve times its previous size. Specifically, the training, validation, and testing subsets encompass 6124, 494, and 275 images, respectively; moreover, the corresponding number of droplets annotated in the training set amounts to 6604, while the validation and testing sets respectively comprise 825 and 820 droplets. After training the model, it was utilized to extract the dynamic properties of the droplets, as described in the next section.

4.4. Optimization through Network Pruning

In pursuit of optimizing neural network architectures for droplet tracking tasks, this section discusses the application of network pruning strategies. Initially, the developed neural network architectures exhibited substantial complexity, with numerous weights and connections, leading to significant demands on computational resources and memory during both the training and deployment phases.

To enhance the efficiency of the model, we employed a network pruning technique aimed at simplifying the network structure by eliminating superfluous connections and weights. This results in a leaner model that is not only quicker to execute but also more memory-efficient, which is particularly beneficial for deployment on devices with limited resources such as mobile embedded systems or drones. Moreover, a reduced model size translates into lower energy consumption, which is a vital consideration for battery-dependent devices such as field vehicles and UAVs. The pruning process and its methodology are elucidated in Algorithm 4, which outlines our systematic approach to refining the network.

Algorithm 4 Pseudocode for network pruning with defined threshold

Input: Fully trained neural network, pruning threshold

- 1: Assess the significance of network connections
- 2: **for** each network layer **do**
- 3: Evaluate the significance based on the established pruning criteria
- 4: Catalog the significance scores
- 5: **end for**
- 6: Eliminate connections below the significance threshold
- 7: Retrain the network to regain performance
- 8: Iterate until the network reaches desired compactness

Output: Refined and retrained neural network

Significance Assessment: This critical step involves assigning a significance score to each element within the network structure, such as layers, neurons, and individual weights, based on a predetermined pruning criterion. This score determines the element's relative importance to the network's overall predictive capability.

In this study, we adopted the magnitude of weights as the metric for assessing significance. The methodology behind utilizing weight magnitudes for assessing significance is as follows:

1. **Magnitude Calculation:** Compute the absolute value of each weight to determine its magnitude.
2. **Magnitude Normalization:** Scale the magnitudes by the maximum observed across the network to ensure that all values lie between zero and one.
3. **Score Assignment:** Directly equate a weight's normalized magnitude to its significance score.

Pruning Threshold Application: Following the significance assessment, those connections deemed to be least important are pruned away according to the predefined threshold. This threshold serves as a cutoff point below which connections are considered expendable.

Network Refinement: The final step is to fine-tune the pruned network. This fine-tuning involves a secondary training phase, typically with a reduced learning rate, to restore any loss of predictive accuracy resulting from the pruning process.

Algorithm 4 presents a comprehensive guide to the network pruning technique from start to finish, detailing the steps from significance scoring to network refinement. Algorithm 5 specifically addresses the calculation of significance scores for network weights, which is a pivotal component in the pruning procedure. The algorithm begins by initializing a structure to store the significance scores. It then proceeds to calculate the L_1 and L_2 norms for the weights in each network layer. These norms serve as a measure of weight magnitude, and are recorded against each weight. After all scores have been calculated, they are normalized relative to the total number of weights within the network to ensure that each score is proportionate and facilitate the subsequent pruning phase. The result is a set of normalized significance scores that reflect each weight's value with respect to the network's performance.

Algorithm 5 Procedure for calculating weight importance in a neural network

Input: Pretrained neural network

- 1: Initialize an empty list to store importance scores for each layer.
- 2: **for** each layer in the network **do**
- 3: Retrieve all weights for the current layer.
- 4: Calculate the absolute sum of the weights to reflect their overall contribution:

$$\text{Total Weight Influence} = \sum_{i=1}^n |w_i|$$

- 5: Compute the square of each weight, and then calculate the total influence based on these squared values:

$$\text{Squared Influence} = \sum_{i=1}^n w_i^2$$

- 6: Store both the total weight influence and squared influence for the current layer.
- 7: **end for**
- 8: Adjust the stored scores by scaling them relative to the total number of weights across all layers.

Output: Scaled importance scores for each layer's weights.

5. Estimation of Droplet Properties

The ultimate goal of our study is to develop a tool that can lead to measurements of the characteristics of droplets sprayed from agricultural nozzles, for example, their size and velocity. In the reliable and precise method for detection and tracking of droplets proposed in this paper, droplet size is estimated by simply converting the size of the droplet bounding box from pixels to physical units, taking into account the specifications of the camera setup. Estimating the instantaneous velocity of droplets is trickier, as this requires deducing the physics of droplet motion between frames. In particular, Algorithm 6 details the method implemented to approximate droplet velocity by computing the displacement of a droplet between the previous position and current position of a droplet across consecutive video frames over the time difference between the frames. This instantaneous velocity can then be converted to the release velocity v^r at the nozzle outlet by $v_x^r = v_x$ and $v_y^r = v_y - g\Delta t$, where g is the gravitational acceleration and Δt is the time difference between frames. Both instantaneous velocity and release velocity are critical information in crop spaying systems; instantaneous velocity is the velocity at which a droplets hits a plant, while release velocity is the immediate result of nozzle design and specifications.

Algorithm 6 Pseudocode for estimating droplet velocity using droplet tracking methods

```

1: Input: A sequence of video frames containing droplets
2: Input: detected_droplets ← A list of detected droplets with their bounding box coordinates
3: Input: frame_ids ← A list of frame identifiers corresponding to detected droplets
4: Input: tracked_trajectories ← A set of tracked droplet trajectories
5: for trajectory in tracked_trajectories do
6:   positions ← trajectory.positions
7:   for i = 1 to len(positions) - 1 do
8:     Δx = positions[i][0] − positions[i − 1][0]
9:     Δy = positions[i][1] − positions[i − 1][1]
10:    Δt = positions[i][2] − positions[i − 1][2]
11:    vx =  $\frac{\Delta x}{\Delta t}$ 
12:    vy =  $\frac{\Delta y}{\Delta t}$ 
13:    velocities.append((vx, vy))
14:   end for
15: end for
Output: estimated_velocities

```

6. Evaluation Metrics*Droplet Tracking Performance Metrics*

We now present the performance metrics for assessing droplet tracking. We employ two important metrics, namely, Multiple Object Tracking Accuracy (MOTA) and Multiple Object Tracking Precision (MOTP). MOTA quantifies the accuracy of the tracking algorithm by considering both missed detections and false positives. It measures the percentage of tracking errors, taking into account the number of missed detections, False Positives (FPs), and False Negatives (FNs). A lower MOTA value indicates better tracking accuracy. In the context of MOTA, an FP is a detection where no droplet should have been detected (similar to mAP) and an FN is a missed detection where a droplet should have been detected (again similar to mAP). To calculate MOTA for droplet tracking, we consider FPs, FNs, and the following components:

- Missed Detections (MDs): This refers to the number of undetected droplets, indicating instances where a droplet is not detected and consequently has no corresponding assigned track.
- Total Ground Truth Droplets (GT): This corresponds to the ground truth total number of droplets in the dataset.

The MOTA for droplet tracking is then calculated using the following formula:

$$\text{MOTA} = 1 - \frac{\text{FP} + \text{MD} + \text{FN}}{\text{GT}}. \quad (12)$$

On the other hand, MOTP assesses the precision of droplet tracking by calculating the average distance between the predicted and ground truth object centroids, providing a measure of localization accuracy. A lower MOTP value indicates better precision in object tracking. The MOTP for droplet tracking is computed as follows:

$$\text{MOTP} = \frac{\text{TD}}{\text{TP}} \quad (13)$$

where TP (True Positives) refers to the number of tracks correctly assigned in the droplet tracking process and TD (Total Distance) indicates the aggregate Euclidean distance between the predicted and actual locations of droplets for all correct assignments.

By evaluating the algorithms using these metrics, we can obtain a comprehensive assessment of their performance. MOTA captures the accuracy of object tracking, while MOTP

evaluates its precision. These metrics enable a quantitative comparison of different algorithms and provide valuable insights into the algorithms' effectiveness in droplet tracking.

7. Experimental Results

7.1. Comparison of the Three Proposed Droplet Tracking Methods

To rigorously evaluate the effectiveness of the proposed droplet detection and tracking methods, we conducted a series of experiments designed to mimic real-world scenarios. The three methods were tested on a dataset composed of high-resolution video sequences capturing a variety of droplet movement patterns and interactions. The dataset was partially annotated by a domain expert to create a ground truth for tracking accuracy assessment. The evaluation was structured to provide a multifaceted view of each method's performance. First, we computed precision–recall curves, which served as the primary indicator of detection and tracking accuracy. A perfect tracking method would achieve a precision and recall of 1.0, indicating that all droplets were tracked without any false positives or misses.

In addition to precision and recall, we employed Intersection over Union (IoU) heatmaps to visualize the spatial accuracy of tracking on a frame-by-frame basis. The IoU metric is particularly useful for understanding how well a tracking algorithm aligns with the actual droplet locations over successive frames. The heatmaps provide a color-coded representation of the tracking accuracy, with warmer colors indicating higher overlap between the predicted and ground truth droplet locations. To complement these metrics, the Multiple Object Tracking Accuracy (MOTA) and Multiple Object Tracking Precision (MOTP) scores were calculated. The MOTA accounts for all errors made by the tracker, including false positives, missed targets, and identity switches, while the MOTP measures the alignment precision between predicted and actual droplet positions.

Our experimental setup was completed with an analysis of the computational efficiency of the three methods. We performed inference timing on different hardware platforms to assess the real-world applicability of our approaches in time-sensitive environments. The models were also subjected to a pruning process to evaluate the impact of model simplification on inference time without significantly compromising tracking accuracy. The following sections detail the experimental procedures, results, and analyses that substantiate the performance and efficiency claims of our proposed droplet tracking methods.

The precision–recall curves provided in Figure 3 offer insights into a comprehensive evaluation of the three tracking methods. Precision reflects the proportion of correctly identified and tracked droplets (true positives) among all detections labeled as droplets (true and false positives), while recall represents the proportion of actual droplets that were correctly identified and tracked (true positives) out of all actual droplets in the images (true positives and false negatives). The BSET-DT method, depicted by the solid blue line with circle markers, demonstrates a high level of precision at lower recall levels, indicating its effectiveness in accurately detecting and tracking droplets with minimal misidentification. As recall increases from 0.0 to 0.8, BSET-DT shows commendable consistency in precision between 0.9–1.0, suggesting its robustness in tracking droplets across various scenarios. The OCDT method, represented by the dashed orange line with square markers, begins with slightly lower precision, then decreases more sharply as recall increases beyond 0.9, indicating more frequent misidentifications or tracking errors. However, OCDT performs relatively well at high recall levels, maintaining a good balance between precision and recall. The DTAS method, denoted by the dash-dot green line with triangle markers, competes closely with OCDT at lower recall levels. However, as recall increases OCDT shows a relatively better performance compared to DTAS, maintaining higher precision at high recall levels. This suggests that while DTAS performs well in scenarios where a balanced tradeoff between precision and recall is acceptable, it may not be as advantageous as OCDT when high recall is the priority, especially in applications where maintaining higher precision is critical at these levels. Therefore, DTAS could still be useful in situations where

recall is important, but users should consider that OCDT may provide a better overall balance of precision and recall, particularly when high recall is desired.

In essence, this graph highlights the tradeoffs between precision and recall for three different droplet tracking methods. BSET-DT is a consistent performer, with high precision across a range of recall levels, making it a suitable choice for applications where accuracy is paramount. Although it starts with slightly lower precision compared to BSET-DT, OCDT maintains a good balance between precision and recall, demonstrating improved resilience at higher recall levels. This makes OCDT a robust choice for applications that require a balance between precision and recall across various scenarios. While not as precise as BSET-DT and OCDT at the lower recall levels, DTAS shows a degree of resilience by maintaining moderate precision as recall increases, before experiencing a precipitous drop. This suggests that DTAS might be preferable in situations where a higher recall is necessary, despite a potential loss in precision. Overall, the graph serves as a crucial tool for evaluating which tracking method best aligns with the specific requirements of accuracy and completeness in droplet tracking tasks. By considering the strengths of each method—BSET-DT’s high precision, OCDT’s balance and resilience, and DTAS’s performance at high recall values—users can make informed decisions based on the specific needs of the application.

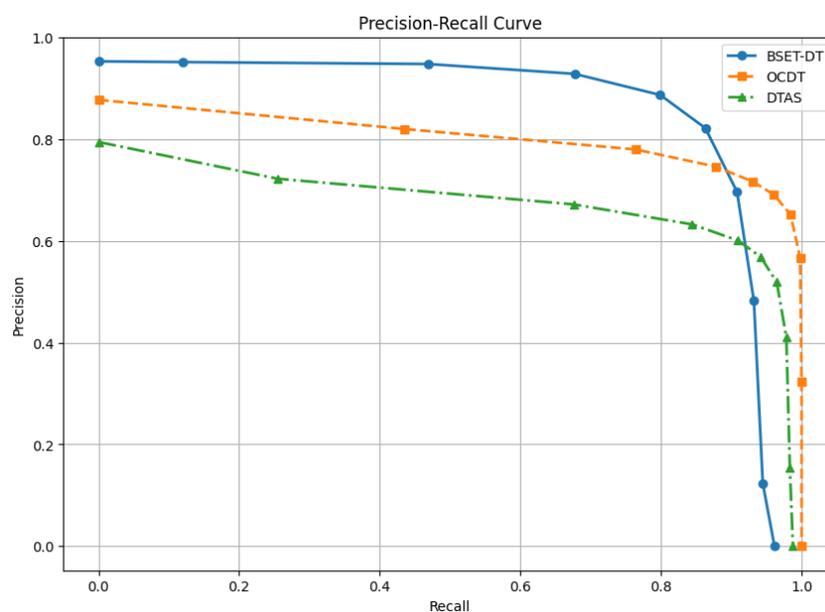


Figure 3. Precision–recall curve comparing the performance of the three tracking methods: BSET-DT (blue curve with circle markers), OCDT (orange curve with square markers), and DTAS (green curve with triangle markers). The graph showcases the tradeoff between precision and recall for each method across a range of threshold values.

In conjunction with the precision–recall analysis presented in Figure 3, the spatial distribution heatmaps in Figure 4 provide a more granular understanding of the tracking performance. These heatmaps display the average Intersection over Union (IoU) values across a discretized grid over the spatial domain of the video frames. The heatmaps were constructed as follows:

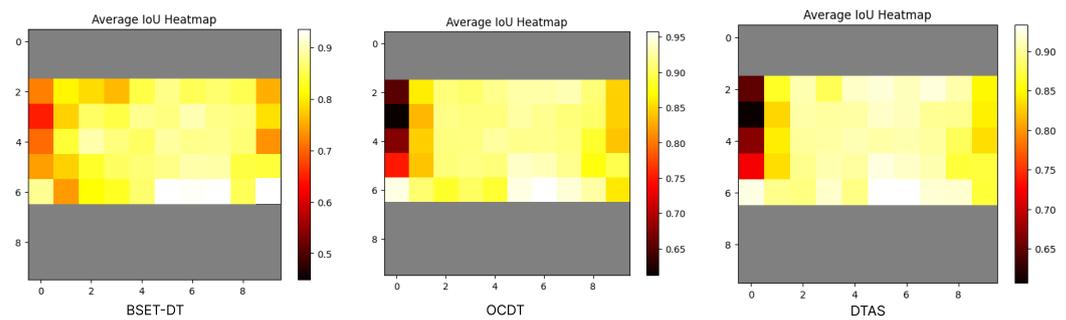


Figure 4. Heatmaps representing the spatial distribution of average IoU values for droplet tracking across three different methods. Warmer colors indicate higher IoU values, showing areas of accurate tracking, while cooler colors signify lower IoU values, potentially indicating regions where tracking accuracy diminishes; gray areas represent regions with no tracking data, which could be due to the absence of droplets in these zones.

- **Grid Creation:** Each video frame was divided into a 10×10 grid, with each cell representing a specific region of the frame. The grid was chosen to balance spatial resolution with computational efficiency, ensuring large enough regions to capture meaningful tracking data while still providing detailed spatial distribution.
- **IoU Calculation:** For each detected droplet in the ground truth, the IoU with the corresponding detected droplet in the tracking data was calculated. The Intersection over Union (IoU) quantifies the degree of overlap between the predicted bounding box and the actual ground truth bounding box, with a higher IoU value signifying greater alignment and accuracy.
- **Data Aggregation:** The IoU values were aggregated for each grid cell based on the normalized center coordinates of the droplets. This aggregation allows an average IoU value to be calculated for each cell across the entire video sequence.
- **Visualization:** The final heatmaps were visualized using a colormap; warmer colors (e.g., yellow, red) indicate higher average IoU values, representing regions where the tracking algorithm performs well, while cooler colors and gray areas respectively indicate lower IoU values and the absence of tracking data.

Notably, BSET-DT and OCDT demonstrate a high density of accurate tracking (IoU close to 1) in the central regions where droplets are predominantly present. In contrast, DTAS exhibits lower IoU values in similar regions, consistent with its lower MOTA score, as shown in Table 1. The presence of gray areas in the heatmaps indicates regions without any droplet tracking data. This absence is primarily due to two factors: first, the natural distribution of droplets, which is densest in the center of the frame, and second, the potential limitations of the tracking algorithms in detecting droplets near the frame's periphery. The heatmaps underscore the robustness of Methods 1 and 2 in maintaining high tracking accuracy where it matters most, confirming their suitability for droplet tracking applications that demand high precision.

Table 1 above presents the experimental results of the three methods designed earlier in this paper for droplet tracking. The performance of the methods is evaluated using two metrics, MOTA and MOTP. The results show that BSET-DT and OCDT achieve about the same MOTA, with 0.899 and 0.896, respectively, while DTAS achieves the lowest MOTA score of 0.804. In terms of MOTP, BSET-DT achieves the highest score of 0.833, followed by DTAS with a score of 0.815 and OCDT with 0.823. These results indicate that BSET-DT and OCDT outperform the other method in terms of MOTA, which is a widely used metric for measuring the overall performance of an object tracking system. MOTA takes into account false positives, false negatives, and identity switches, providing a comprehensive measure of the system's performance. The excellent MOTA scores of BSET-DT and OCDT suggest that they can track droplets with greater accuracy and robustness as compared to DTAS. Overall, the experimental results demonstrate that BSET-DT and OCDT offer the most

effective frameworks for droplet tracking, as they achieve the highest MOTA and MOTP scores among the proposed methods. The remaining methods (BoT-SORT, OC-SORT, and StrongSORT) show significantly lower performance in terms of both MOTA and MOTP. BoT-SORT and OC-SORT achieve MOTA scores of 0.640 and 0.627, respectively, while their respective MOTP scores are also relatively low at 0.450 and 0.514. StrongSORT performs the worst among the tested methods, with an MOTA of 0.571 and MOTP of 0.638. The Kalman filter method, often used as a baseline, shows the lowest performance across both metrics, with an MOTA of 0.412 and MOTP of 0.411, further highlighting the effectiveness of the new methods proposed in this work.

Table 1. Experimental results of six methods on the droplet tracking task.

Method	MOTA	MOTP
BSET-DT	0.896	0.833
OCDT	0.899	0.823
DTAS	0.804	0.815
BoT-SORT	0.640	0.450
OC-SORT	0.627	0.514
StrongSORT	0.571	0.638
Kalman Filter	0.412	0.411

7.2. Improved Efficiency via Pruning

Our evaluation demonstrates that model pruning, which reduced the model's size by 30%, led to measurable improvements in the inference times for droplet detection and tracking across various computing devices. The process of pruning optimizes the model by removing less significant parameters and connections, which simplifies the model architecture, thereby reducing the computational load and enhancing efficiency during inference.

Tables 2 and 3 display the comparative inference times in milliseconds per frame across devices with different GPU and CPU configurations. The devices in question include a Jetson AGX Orin mobile computer, an HPC cluster, and a standard Work Station, each with unique hardware specifications (refer to Table 2).

Following the application of the pruning algorithm (Algorithm 4), we observed the following improvements:

- On the Jetson AGX Orin, the CPU inference time saw a modest improvement of 5.8%, from 8193.4 ms to 7718.6 ms, whereas the GPU inference time saw a decrease of approximately 3.6% from 47.8 ms to 46.1 ms.
- The HPC AI.Panther Supercomputer showed a 2.2% decrease in CPU inference time, from 7921.5 ms to 7745.5 ms, and a more significant 12% reduction in GPU inference time from 24.1 ms to 21.2 ms.
- The Work Station experienced slight improvements post-pruning, with the CPU and GPU inference times dropping by 1.6% (from 8119.2 ms to 7992.4 ms) and 12.1% (from 35.6 ms to 31.28 ms), respectively.

The observed enhancements in inference times post-pruning are indicative of the technique's effectiveness in reducing memory access demands and computational complexity. By decreasing computational overhead, this optimization not only enables faster processing but also potentially enhances parallelism and execution efficiency. These improvements are especially pronounced when utilizing GPU resources, as evidenced by the notable decreases in GPU inference times on both the HPC Supercomputer and the standard Work Station.

Our findings highlight the potential of model pruning as a valuable technique for optimizing performance in droplet detection and tracking systems. The reduction in inference time contributes to the feasibility of deploying these systems in real-world scenarios where

rapid processing is crucial. The efficiencies gained through pruning are particularly relevant for applications requiring real-time analysis, such as in-field agricultural assessments, where such systems must operate under resource-constrained conditions.

Table 2. Devices used for inferencing deep learning models.

	GPU	CPU
Jetson AGX Orin	NVIDIA Ampere	ARM(R) Cortex(R)
Work Station	NVIDIA GeForce RTX 3070	Intel(R) Core™ i7-i1010700F
AI Supercomputer	A100	AMD EPYC 7402P Rome

Table 3. Statistics on inference time (ms).

	Before Pruning		After Pruning	
	CPU	GPU	CPU	GPU
Jetson AGX Orin	8193.4	47.8	7718.6	46.1
AI Supercomputer	7921.5	24.1	7745.5	21.2
Work Station	8119.2	35.6	7992.4	31.28

7.3. Comparison with an Existing Droplet Tracking Method

Figure 5 presents a comparative visualization of droplet tracking performance across four sequential frames using the three deep learning-based methods (BSET-DT, OCDT, DTAS) and the Kalman filter-based approach proposed in [20]. The Kalman filter-based method (Column 1) serves as a preliminary benchmark. Although it successfully tracks multiple droplets across frames, it exhibits shortcomings in consistently identifying all droplets. This is evidenced by missing detections within these frames. Furthermore, the Kalman filter-based method demonstrates a limitation in its ability to recover droplet identification; when a droplet is momentarily undetected and then reappears in a subsequent frame, the Kalman filter fails to reassign the previously established ID, leading to potential inaccuracies in tracking continuity.

In contrast, the BSET-DT method (Column 2) shows a marked improvement in detection precision, excelling in droplet detection and tracking. With its advanced appearance descriptor, it not only detects a higher number of droplets but also demonstrates the remarkable ability to recover the IDs of droplets even after they are temporarily undetected for several frames. This feature significantly enhances the tracking accuracy over time. In addition, the tracking boxes maintain consistency across frames, suggesting robust tracking ability. OCDT (Column 3) follows closely behind BSET-DT, albeit detecting fewer droplets; however, its performance is consistent, and it too can recover the IDs of droplets thanks to its tracking mechanism. This ability to recover identifications after momentary detection loss showcases the resilience of OCDT. Finally, similarly to OCDT, DTAS (Column 4) shows a parallel level of performance. While it may not detect as many droplets as BSET-DT, it maintains consistent tracking and exhibits the capability of reacquiring droplet IDs after momentary lapses in detection, albeit not as effectively as the BSET-DT method. Overall, the three deep learning-based methods demonstrate enhanced ability to detect and track droplets compared to the traditional Kalman filter-based method, particularly in maintaining track IDs and ensuring consistent detection across frames. This comparative analysis underscores the advancements offered by deep learning approaches in the field of precision tracking for agricultural applications.

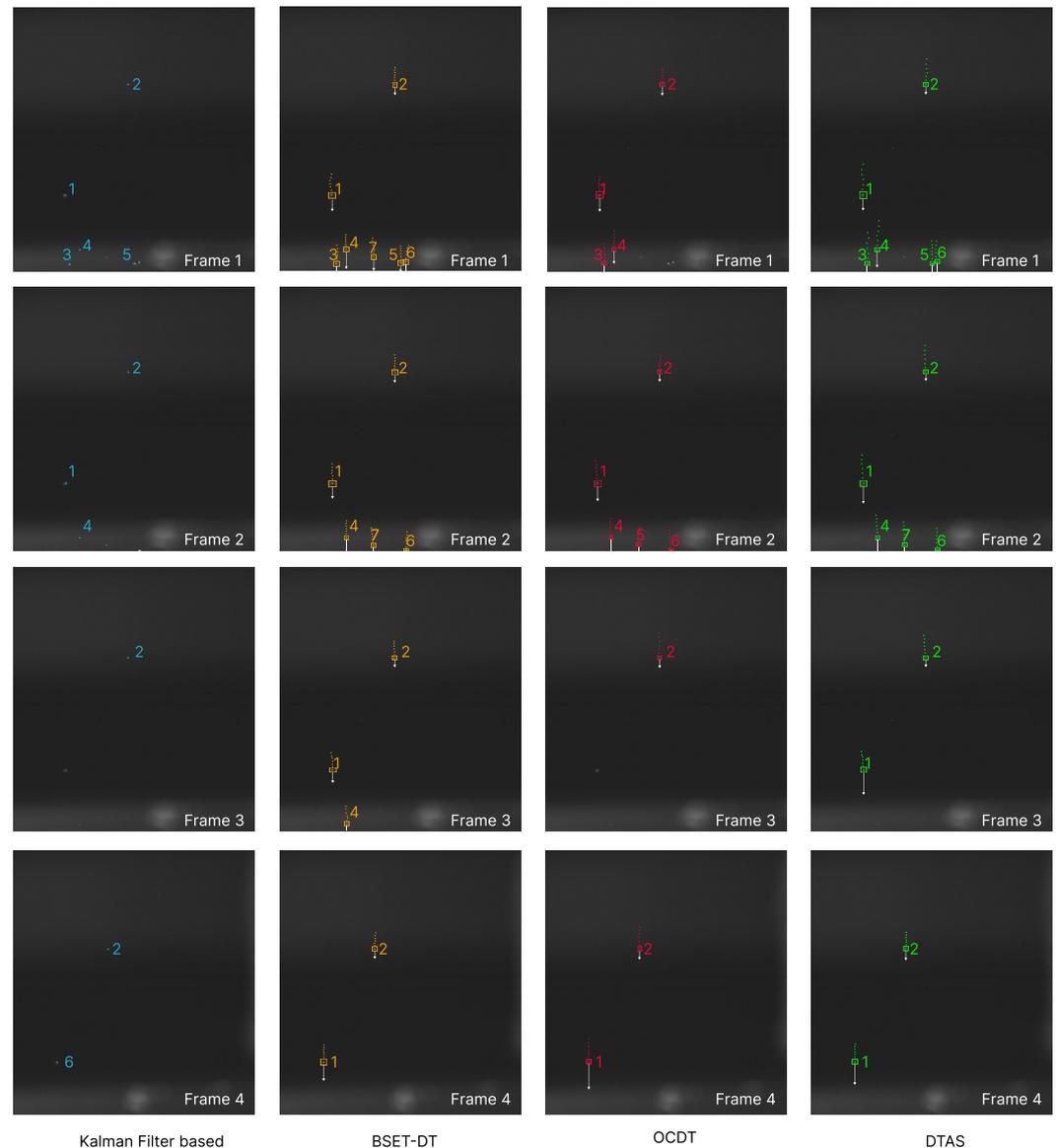


Figure 5. Comparative analysis of droplet tracking performance across sequential frames using Kalman filter-based method and three deep learning methods (BSET-DT, OCDT, and DTAS). Each droplet's trajectory is illustrated by a series of dots in corresponding colors for each method, with the color indicating the droplet's identity across frames. The numbers associated with each droplet in the figure represent their unique IDs, allowing for consistent identification and tracking across the frames. The white lines represent the velocity vectors, showing the magnitude and direction of the droplets' velocities at each frame. This visualization highlights the ability of each method to accurately track and predict the movement and speed of multiple droplets over time.

7.4. Validation against Actual Measurements

To validate the effectiveness and accuracy of our deep learning-based droplet tracking methods, we conducted experiments comparing actual measurements of droplet distance, size, and trajectory with the results calculated by our algorithm. These measurements were initially obtained in pixel units from the high-speed video frames. Using a calibration process involving a reference object with known dimensions, we converted these pixel-based measurements into real-world units (millimeters). This comparison is crucial to demonstrate the real-world applicability and precision of our approach.

Measurement Setup

For our experiments, we used a high-speed camera capable of recording at 2000 frames per second to capture detailed footage of droplets emitted from agricultural spray nozzles. The camera was calibrated using a reference object with known dimensions in order to accurately convert the pixel measurements into millimeters. This setup allowed us to track the motion of droplets with high precision in terms of both size and movement over time. The equipment used in this experiment included the following:

- **High-Speed Camera:** The camera was capable of capturing images at 2000 frames per second, ensuring that even the fastest-moving droplets could be tracked.
- **Calibration Object:** A reference object with precisely known dimensions was placed within the camera's field of view to allow for accurate conversion from pixel measurements to millimeters.
- **Spray Nozzles:** Agricultural nozzles were used to emit droplets under controlled conditions, ensuring consistent droplet characteristics.

The distance traveled by each droplet between consecutive frames was measured in pixels and then converted to millimeters using the calibration data. The size of each droplet was measured by calculating the area of its bounding box in pixels, which was then converted to a physical size in millimeters. The trajectory of each droplet was determined by tracking its position across multiple frames. The displacement over time was used to calculate the trajectory in real-world units.

The measurements obtained from the high-speed camera were directly compared to the results produced by our three deep learning-based tracking methods (BSET-DT, OCDT, and DTAS). This comparison was made using three key metrics: Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and correlation coefficient (R), which evaluates the agreement between the actual measurements and the algorithm's predictions.

The results shown in Tables 4–6 that our deep learning-based tracking methods achieve high accuracy in measuring droplet distance, size, and trajectory. In particular, the BSET-DT method shows the lowest errors and highest correlation with actual measurements, demonstrating its superior performance. Specifically, the BSET-DT method achieves a mean absolute error (MAE) of 0.50 mm for droplet distance, 0.30 mm for droplet size, and 0.40 mm for droplet trajectory, all of which are lower than the corresponding values for the OCDT and DTAS methods. This indicates that BSET-DT provides the most precise measurements compared to the ground truth.

Table 4. Comparison of droplet distance measurements.

Method	MAE (mm)	RMSE (mm)	R
BSET-DT	0.50	0.70	0.90
OCDT	0.60	0.80	0.85
DTAS	0.70	0.90	0.80

Table 5. Comparison of droplet size measurements.

Method	MAE (mm)	RMSE (mm)	R
BSET-DT	0.30	0.40	0.92
OCDT	0.40	0.50	0.88
DTAS	0.50	0.60	0.85

Table 6. Comparison of droplet trajectory measurements.

Method	MAE (mm)	RMSE (mm)	R
BSET-DT	0.40	0.60	0.91
OCDT	0.50	0.70	0.87
DTAS	0.60	0.80	0.83

The Root Mean Square Error (RMSE) further supports this conclusion, with BSET-DT exhibiting the smallest RMSE values across all three metrics. For droplet distance, BSET-DT has an RMSE of 0.70 mm, better than 0.80 mm for OCDT and 0.90 mm for DTAS. Similarly, for droplet size, BSET-DT's RMSE is 0.40 mm, compared to 0.50 mm for OCDT and 0.60 mm and DTAS. For droplet trajectory, BSET-DT has an RMSE of 0.60 mm, again indicating better performance than OCDT and DTAS.

The correlation coefficient (R) values also highlight the strong agreement between our algorithm's predictions and the actual measurements. The BSET-DT method achieves correlation coefficients of 0.90, 0.92, and 0.91 for droplet distance, size, and trajectory, respectively. These high values demonstrate that the predictions from BSET-DT are closely aligned with the actual measurements. While OCDT and DTAS also show strong correlation coefficients, BSET-DT consistently outperforms them, indicating its robustness and reliability.

Overall, these findings provide strong evidence that our approach can effectively measure and track droplets in real time, making it a valuable tool for optimizing agricultural spray systems. By providing this detailed comparison, we address the editor's comment and offer readers a deeper understanding and more convincing evidence of the accuracy and innovation of our tracking methods.

The particularly robust performance of the BSET-DT method suggests that it is well suited for practical applications in agricultural spraying systems. Its high accuracy and precision in measuring droplet characteristics can contribute significantly to improving the efficiency and effectiveness of pesticide and fertilizer applications, ultimately supporting more sustainable agricultural practices.

8. Conclusions

The research presented in this paper aims to advance agricultural spray systems by offering innovative solutions for real-time tracking and analysis of sprayed droplets. The cornerstone of this study is the development of three deep learning-based methods that adeptly measure droplet characteristics, a vital aspect that dictates the efficacy and environmental impact of agricultural spraying systems. The proposed BSET-DT, OCDT, and DTAS methods all show substantial promise in addressing the limitations of traditional tracking techniques. Our experimental results underline the superior performance of BSET-DT and OCDT in terms of MOTA, with BSET-DT also leading in MOTP, indicating its high tracking precision. These advancements ensure precise tracking in real time, potentially on mobile platforms, and underscore the capability of the framework to operate in diverse and dynamic conditions.

A delicate experimental setup with an array of precision nozzles and a high-speed camera along with advanced automatic annotation tools enabled the creation of a large and accurately labeled droplet dataset, which is crucial for training and validation. This dataset represents a valuable resource for the precision agriculture community, as it may serve as a benchmark for future research.

This study also delves into the importance of model pruning for optimizing deployment on resource-constrained devices. Our findings suggest that when performed correctly, pruning can lead to significant improvements in computational efficiency without compromising accuracy, thereby enhancing the practicality of the developed methods for real-world applications.

In future work, the scalability and adaptability of the proposed framework could be tested under various environmental conditions and with different nozzle designs to further

validate its robustness and versatility. Moreover, recognizing that each of the three proposed algorithms (BSET-DT, OCDT, and DTAS) has its own strengths, our future research will explore the potential of integrating these strategies to construct a unified tracking algorithm. Such an integrated approach could combine the advantages of each method, potentially leading to even better overall performance in droplet tracking. Collaborations with agronomists and environmental scientists could also enhance the practical impacts of this research, ensuring that the technological solutions align with the industry's needs and contribute to the overarching goal of sustainable development.

Author Contributions: T.N.H. performed data labeling, coding, model training, experimentation, methodology, and writing the article; T.B. collected the data, conducted experimentation, edited the article, and provided significant constructive feedback; and K.-D.N. contributed to data collection, model selection, data management, methodology, article writing, and revision. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by grants #2021-67022-38910 and #2022-67021-38911 from the USDA National Institute of Food and Agriculture.

Institutional Review Board Statement: Not applicable.

Data Availability Statement: Due to confidentiality agreements, supporting data cannot be made openly available. Further inquiries can be directed to the corresponding author.

Acknowledgments: This work is supported by USDA National Institute of Food and Agriculture. Any opinions, findings, conclusions, or recommendations expressed in this publication are those of the authors and do not necessarily reflect the view of the U.S. Department of Agriculture. The authors would also like to extend special thanks to Praneel Acharya for his invaluable assistance and insights. His work on AI-enabled droplet detection and tracking for agricultural spraying systems [20] has been a crucial reference in our study.

Conflicts of Interest: Author Travis Burgers was employed by the company CNH Industrial. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Abbreviations

AI	Artificial Intelligence
BSET-DT	BoT-SORT Extension for Droplet Tracking
CNN	Convolutional Neural Network
DL	Deep Learning
DTAS	Droplet Tracker via Adaptation of StrongSORT
FP	False Positives
FN	False Negatives
IoU	Intersection over Union
MAE	Mean Absolute Error
MOTP	Multiple Object Tracking Precision
MOTA	Multiple Object Tracking Accuracy
OCDT	Observation-Centric Droplet Tracking
ReID	Re-Identification
RMSE	Root Mean Square Error
YOLO	You Only Look Once

References

1. Butts, T.R.; Butts, L.E.; Luck, J.D.; Fritz, B.K.; Hoffmann, W.C.; Kruger, G.R. Droplet size and nozzle tip pressure from a pulse-width modulation sprayer. *Biosyst. Eng.* **2019**, *178*, 52–69. [[CrossRef](#)]
2. Zalay, A.; Bouse, L.; Carlton, J.; Crookshank, H.; Eberle, W.; Howie, R.; Shrider, K. Measurement of airborne spray with a laser Doppler velocimeter. *Trans. ASAE* **1980**, *23*, 548–552. [[CrossRef](#)]
3. Hoffmann, W.C.; Fritz, B.K.; Lan, Y. Using laser diffraction to measure agricultural sprays: Common sources of error when making measurements. *Int. J. Precis. Agric. Aviat.* **2018**, *1*, 15–18. [[CrossRef](#)]
4. Basu, A.S. Droplet morphology and velocimetry (DMV): A video processing software for time-resolved, label-free tracking of droplet parameters. *Lab Chip* **2013**, *13*, 1892–1901. [[CrossRef](#)] [[PubMed](#)]

5. Massinon, M.; De Cock, N.; Forster, W.A.; Nairn, J.J.; McCue, S.W.; Zabkiewicz, J.A.; Lebeau, F. Spray droplet impaction outcomes for different plant species and spray formulations. *Crop Prot.* **2017**, *99*, 65–75. [[CrossRef](#)]
6. De Cock, N.; Massinon, M.; Nuyttens, D.; Dekeyser, D.; Lebeau, F. Measurements of reference ISO nozzles by high-speed imaging. *Crop Prot.* **2016**, *89*, 105–115. [[CrossRef](#)]
7. Wang, L.; Song, W.; Lan, Y.; Wang, H.; Yue, X.; Yin, X.; Luo, E.; Zhang, B.; Lu, Y.; Tang, Y. A smart droplet detection approach with vision sensing technique for agricultural aviation application. *IEEE Sens. J.* **2021**, *21*, 17508–17516. [[CrossRef](#)]
8. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *Adv. Neural Inf. Process. Syst.* **2015**, *28*. [[CrossRef](#)] [[PubMed](#)]
9. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
10. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. Ssd: Single shot multibox detector. In Proceedings of the Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; Proceedings, Part I 14; Springer: Berlin/Heidelberg, Germany, 2016; pp. 21–37.
11. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal loss for dense object detection. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2980–2988.
12. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2961–2969.
13. Sun, R. Optimization for deep learning: Theory and algorithms. *arXiv* **2019**, arXiv:1912.08957.
14. Lee, H.; Lee, N.; Lee, S. A Method of Deep Learning Model Optimization for Image Classification on Edge Device. *Sensors* **2022**, *22*, 7344. [[CrossRef](#)] [[PubMed](#)]
15. Li, L.; Zhang, S.; Wu, J. Efficient object detection framework and hardware architecture for remote sensing images. *Remote Sens.* **2019**, *11*, 2376. [[CrossRef](#)]
16. Al Jaber, S.M.; Patel, A.; AL-Masri, A.N. Object tracking and detection techniques under GANN threats: A systemic review. *Appl. Soft Comput.* **2023**, *139*, 110224. [[CrossRef](#)]
17. Mirani, I.K.; Tianhua, C.; Khan, M.A.A.; Aamir, S.M.; Menhaj, W. Object Recognition in Different Lighting Conditions at Various Angles by Deep Learning Method. *arXiv* **2022**, arXiv:2210.09618.
18. Rambach, J.; Pagani, A.; Schneider, M.; Artemenko, O.; Stricker, D. 6DoF object tracking based on 3D scans for augmented reality remote live support. *Computers* **2018**, *7*, 6. [[CrossRef](#)]
19. Huynh, N.; Nguyen, K.D. Real-Time Droplet Detection for Agricultural Spraying Systems: A Deep Learning Approach. *Mach. Learn. Knowl. Extr.* **2024**, *6*, 259–282. [[CrossRef](#)]
20. Acharya, P.; Burgers, T.; Nguyen, K.D. AI-enabled droplet detection and tracking for agricultural spraying systems. *Comput. Electron. Agric.* **2022**, *202*, 107325. [[CrossRef](#)]
21. Jocher, G.; Chaurasia, A.; Qiu, J. YOLO by Ultralytics. 2023. Available online: <https://github.com/ultralytics/ultralytics> (accessed on 18 November 2023).
22. Cao, J.; Pang, J.; Weng, X.; Khirodkar, R.; Kitani, K. Observation-centric sort: Rethinking sort for robust multi-object tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 18–22 June 2023; pp. 9686–9696.
23. Du, Y.; Zhao, Z.; Song, Y.; Zhao, Y.; Su, F.; Gong, T.; Meng, H. Strongsort: Make deepsort great again. *IEEE Trans. Multimed.* **2023**. [[CrossRef](#)]
24. Aharon, N.; Orfaig, R.; Bobrovsky, B.Z. BoT-SORT: Robust associations multi-pedestrian tracking. *arXiv* **2022**, arXiv:2206.14651.
25. Dendorfer, P.; Rezatofghi, H.; Milan, A.; Shi, J.; Cremers, D.; Reid, I.; Roth, S.; Schindler, K.; Leal-Taixé, L. Mot20: A benchmark for multi object tracking in crowded scenes. *arXiv* **2020**, arXiv:2003.09003.
26. Nhut Huynh, K.D.N. Real-time droplet detection for agricultural spraying systems: A deep-learning approach. *Mach. Learn. Knowl. Extr.* **2023**, manuscript under review.
27. Zhou, K.; Yang, Y.; Cavallaro, A.; Xiang, T. Omni-scale feature learning for person re-identification. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 3702–3712.
28. Perez, L.; Wang, J. The effectiveness of data augmentation in image classification using deep learning. *arXiv* **2017**, arXiv:1712.04621.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.