

results were obtained in the first year of the HALY.ID project, we do not discuss them here. Instead, we focus on the computer vision algorithms used to detect BMSB in images collected autonomously by the drone. From this perspective, it is evident that the majority of achievements related to BMSB detection have been generated within the HALY.ID project. Furthermore, unlike others, we are currently exploring solutions that can accurately determine the BMSB's position directly from aerial or in-field images.

### B. Puncture Detection

In literature, efforts specifically concerning the detection of insect punctures on harvested fruits refer to visible superficial damage on the skin [21], generally detectable in the Vis-NIR range (400–1000 nm). Typically, the main applications for detecting nonvisible damage, internal to the fruit pulp, using SWIR-HSI, refer to the presence of bruising [22], [23], [24]. Efficient techniques for rapid and effective quality assessment of fruits and vegetables are essential to meet the growing consumer demand for better, more consistent, and safer food products. Lu et al. [25] focused on three innovative HSI-based techniques or sensing platforms—spectral scattering, integrated reflectance and transmittance, and spatially resolved spectroscopy—that have been developed in the laboratory for property and quality evaluation of fruits, vegetables, and other food products. Similarly, Alamar et al. [26] focused on the detection of mechanical damages, including bruising. Interestingly, Zamljen et al. [27] examined the metabolic response of apples to BMSB punctures. Compared to healthy fruits, the sugar and phenolic compounds contents differ, suggesting that NIR-HSI imaging could be useful in detecting damages caused by the BMSB. To the best of the authors' knowledge, there are no previous works concerning BMSB puncture detection and its annotation, which followed a practical on-field protocol and involved numerous samples collected over several years.

## III. BMSB DETECTION USING RGB IMAGING

To develop an intelligent system for insect detection in orchards using images and CNNs, the first step is to collect a database of relevant images of insects in the trees, which need to be labeled and categorized to identify the type of insects encountered [28]. The next step is image preprocessing, which involves removing noise and distortion, resizing images, adjusting brightness and contrast, and data augmentation. From the preprocessed dataset, a representative subset of images needs to be selected for training CNNs to identify and classify insects in the images and adjust internal weights to improve accuracy. Then, the CNN must be validated and tested on separate image subsets to evaluate its performance.

Within HALY.ID, the following NNs were used to identify the BMSB insect: YOLOv5, YOLOv9, YOLOv10, SDD with VGG16 backbone, and DERT with ResNet-50 backbone.



Fig. 1. Examples of images from the dataset used in this article.

### A. Experimental Setup

Before starting, we created a dataset of RGB images obtained from several UAV flights in real conditions in the study orchard in Carpi (Modena, Italy). Obviously, the larger the database used for training and validating NNs, the higher the performance of the created models can be [29]. Therefore, we augmented the initial image set due to its limited size. The images acquired by the UAV are 24-bit RGB, with dimensions of  $5184 \times 3888$  pixels each (see Fig. 1). Since the BMSB occupies a minuscule fraction of the entire image, it is evident that directly inputting the raw image into the NN would yield unpromising results. This claim is enforced by Betti Sorbelli et al. [20], who attributed the cause of performance degradation to the loss of information resulting from image scaling. Indeed, resizing the images to  $640 \times 640$  pixels from their original sizes ( $5184 \times 3888$  pixels) renders the BMSB practically invisible. Consequently, we processed the images by dividing them into nonoverlapping patches of size  $640 \times 640$  pixels. We divided then the dataset into 70%, 20%, and 10% for training, validation, and test sets, respectively, by automatically and randomly distributing the indexed dataset with a script created for this purpose. The dataset contains 402 images and, as a consequence of the slicing technique, is decomposed in 17 766 patches of which 423 are positive ones (with BMSB).

Due to the slicing procedure, the testing results of the proposed models are conducted on two test sets: ALL that means patches with and without BMSB (1920 patches), and ONLY that means patches with BMSB (71 patches).

### B. Experimental Results

To evaluate the network performance, we used metrics, such as precision ( $P$ ), recall ( $R$ ), mean average precision ( $mAP$ ), and intersection over union (IoU), summarized in Table I, where TP, FP, and FN represent true positive, false positive, and false negative, respectively. We also include other indicators that will be used in subsequent evaluations.

The SSD is a single shot detector known for its efficiency and accuracy. Like YOLOv5, SSD performs object detection, namely, object localization and classification, in a single forward pass. Key to its success is the use of a series of convolutional layers that produce feature maps at multiple scales, enabling the SSD to detect objects of various sizes within the image. The implementation of SSD in our application was implemented using the PyTorch library. For the SSD backbone we leverage VGG16 [30]. Although SSD demonstrates a rapid detection