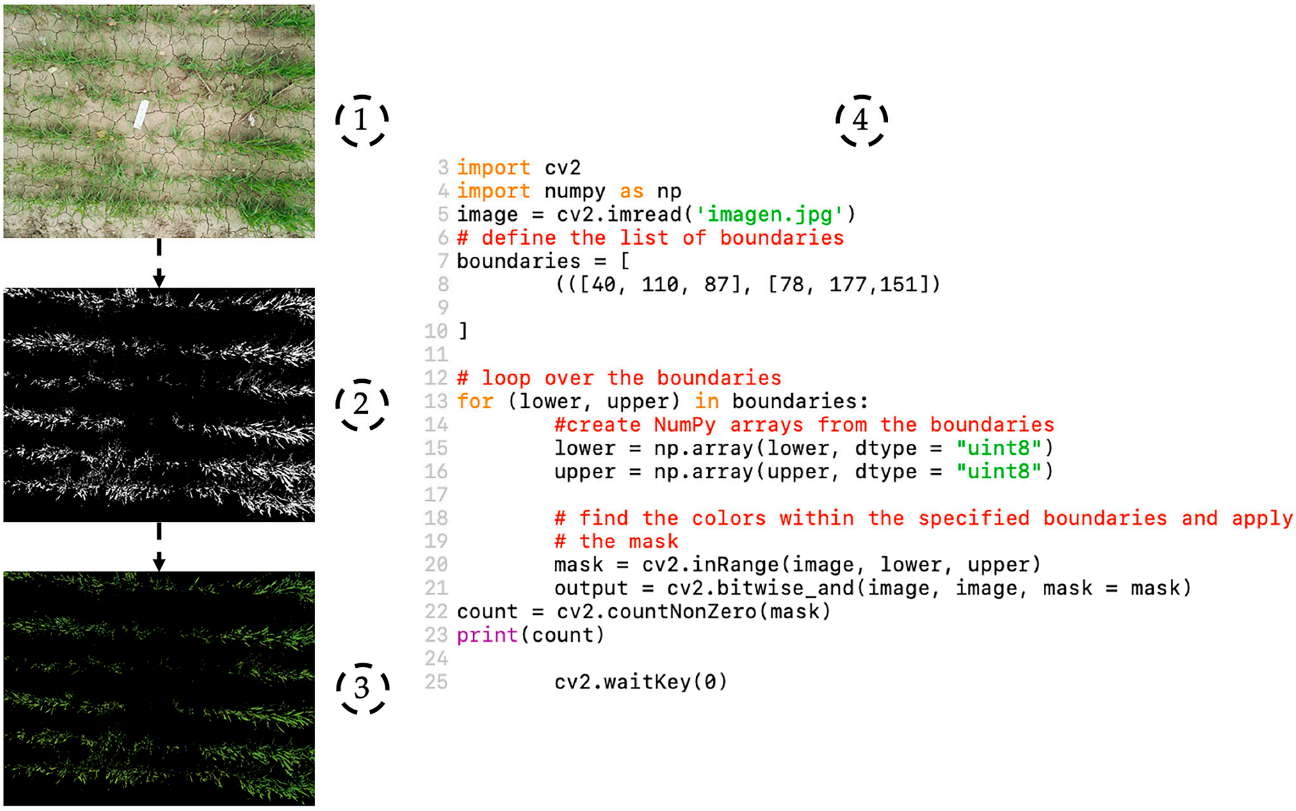
Supplementary Materials

S1 Image Segmentation Method for Nadir-View Images Using a Python Script

The nadir-view images taken with the HTPP were analyzed using a Python script in combination with OpenCV and NumPy libraries. To calculate CC the image was loaded using the OpenCV function *cv2.imread ()*. It is important to realize that OpenCV reads the image in BGR format. Keeping this in mind, a color boundary (lower [40, 110, 87] and upper [78, 177, 151]) with the range of shades of green presents in the images was specified. The OpenCV function *cv2.inRange ()* was used to find the colors with the specified boundaries. A mask image was obtained as an output from this function. Finally, using the mask as input, the OpenCV function *countNonZero ()* was used to count the white (255) pixels (Figure S1). So, knowing the total number of pixels in the image and the pixels counted with the last function, the CC was calculated.



```
import cv2
import numpy as np
image = cv2.imread('imagen.jpg')
# define the list of boundaries
boundaries = [
        (([40, 110, 87], [78, 177,151])

]

# loop over the boundaries
for (lower, upper) in boundaries:
        #create NumPy arrays from the boundaries
        lower = np.array(lower, dtype = "uint8")
        upper = np.array(upper, dtype = "uint8")

        # find the colors within the specified boundaries and apply
        # the mask
        mask = cv2.inRange(image, lower, upper)
        output = cv2.bitwise_and(image, image, mask = mask)
count = cv2.countNonZero(mask)
print(count)

        cv2.waitKey(0)
```

Figure S1. Description of image processing for nadir-view images. In (1) the original RGB image is shown. The mask after finding the colors within the specified boundaries is shown in (2). In (3), the green pixels in the image that have a corresponding white (255) value in the mask are shown. Part of the Python script used to process the images is depicted in (4).

S2 Image Segmentation Method for Hemispherical Images Using Can-Eye Software

Segmentation as a previous step for LAI estimations using hemispherical images was made following instructions indicated by the Can-Eye software. The image taken with the Go Pro is uploaded from the disk using the software (1). Then, using the image dimensions (Length x Width), the optical center of the image is calculated, and the image is processed for the next step (2). In (3), the pixels from the sky are selected to extract the values of vegetation pixels (4). When the selection is finished, the software executes its own algorithms to show a segmented image (5). Finally, after executing additional algorithms an excel file with the LAI values and other parameters are obtained as output (6).

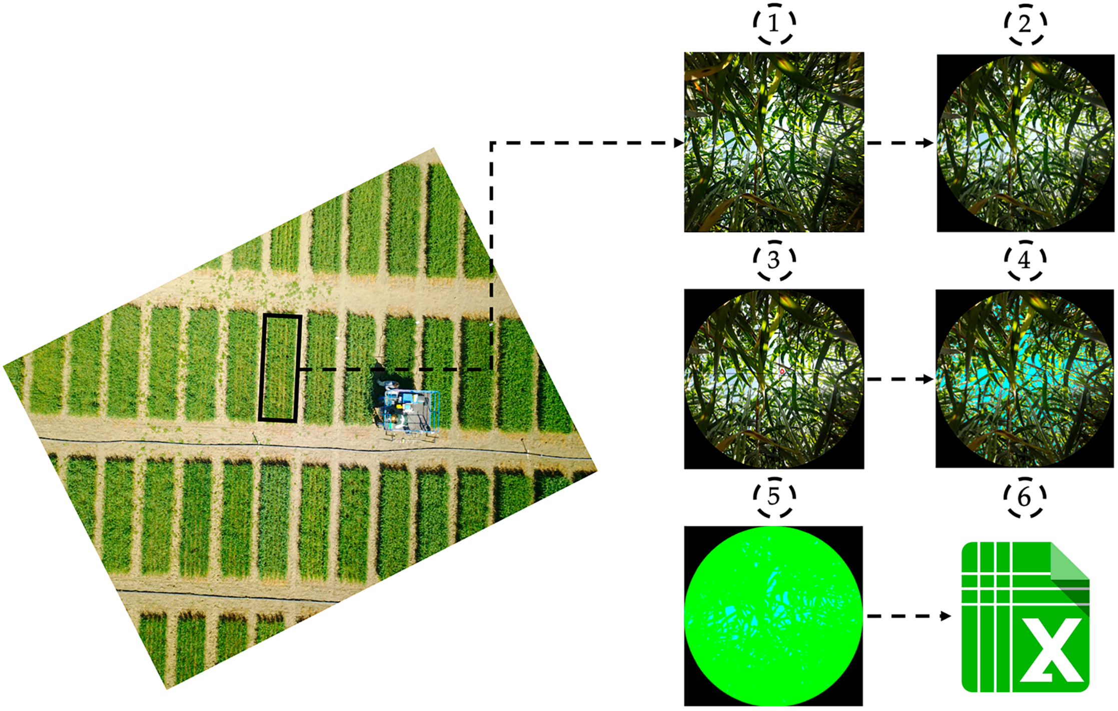


Figure S2. Description of image processing to obtain LAI from hemispherical images and Can-Eye software.