Project 2: Ant Weber Length Estimation using YOLO-Based Deep Learning Framework

Elyes Ben Chaabane, Vivien Gaillet, Amel Abdelraheem

***Abstract*—The automated measurement of morphological features in ants presents a significant challenge in evolutionary biology and entomology research. In this project, we propose a deep learning framework for automatically detecting and measuring Weber's length (thorax length) in ant specimens from the AntWeb database. We propose a multi-component system that combines YOLOv11-based landmark detection with scale bar identification and optical character recognition. Starting with a dataset of 200 annotated ant-profile images, we expanded it to 1,850 images through pseudo-labeling techniques. Our experiments compared small (9.4M parameters) and large (25.3M parameters) YOLO models under various training schemes, including data augmentation and early stopping. The best-performing model achieved a mean average precision (mAP) of 0.650 and a normalized Weber length error of 0.0359. For the scale bar detection, we successfully integrated DeepLSD for line segment detection and EasyOCR for scale value recognition, achieving 96.8% accuracy in scale value detection. Our codes can be accessible Here.**

I. Introduction

Deep learning and computer vision hold great potential for transforming entomology and insect monitoring. For instance, by leveraging sensor-based technologies such as camera traps, radar, and acoustic sensors, researchers can gather vast datasets that offer valuable biological insights [1]. One of the primary objectives of evolutionary biology is to understand patterns of morphological variation and the evolutionary factors that drive them [2]. Specifically in ants, understanding these morphological traits is key to unraveling evolutionary patterns, as it directly influences functional and ecological adaptations within ant populations such as foraging, nesting, and functional roles [3].

A significant challenge in this field is the reliance on manual annotations for extracting morphological features, which is time-consuming and prone to human error. To address this, automated methods for detecting these features are needed. One promising approach is to frame the problem as a landmark detection task, leveraging deep learning techniques, specifically transfer learning to accelerate and enhance detection accuracy. Notably, existing studies have successfully applied transfer learning for landmark detection of other animal features [4], suggesting its potential applicability in ant morphological studies. This project focuses on using deep learning methods for accurate ant morphological features from images. In particular, we propose using a framework based on the latest iteration of the SOTA object detection model YOLOv11 [5] (You-Only-Look-Once) to detect the ant's thorax length (also known as Weber's length) which can enhance our understanding of ant morphology and contribute to broader insect ecology research.

II. Methods

A. Dataset

AntWeb is the world's largest online database of images, specimen records, and natural history information on ants [6]. currently documenting 17,339 valid species and subspecies. It includes a vast collection of 868,119 specimen records contributed by 21 collaborating institutions. Among these, 18,957 species and subspecies have been imaged, covering both valid taxa and those that are indeterminate or morphospecies. The database features detailed visual documentation with 61,097 specimens photographed, amounting to an impressive 260,007 total specimen images.

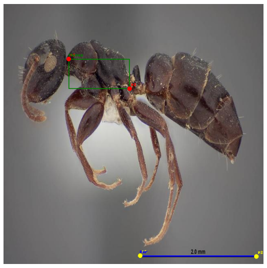


Figure 1. Ant-profile view showing thorax (in red) and scale bar (in yellow) keypoints with scale value directly above the bar (i.e 2mm)

Initially, we were provided with a subset of 200 annotated ant-profile images, randomly sampled from the AntWeb database. Each image included annotations for:

- Two points defining Weber's length keypoints.
- Two points defining the image's scale bar.
- The scale value, measured in millimeters (mm), as shown in Figure 1.

The images were high-resolution with varying dimensions. To maintain consistency with YOLO's pretraining, all images were resized to 640×640 pixels. Subsequently, we normalized the landmarks to the range [0, 1] and defined bounding boxes based on each landmark pair. To support experimentation, three versions of the dataset were created: Thorax landmarks only, Scale landmarks only and Both thorax and scale landmarks.

1) Dataset Enhancement: Pseudo-Labeling: To expand the thorax dataset, a YOLOv11 model was initially trained on the 200 human-annotated images. This trained model was then used to predict landmarks on the remaining unlabeled images. Predicted annotations were manually reviewed, resulting in the addition of 1,850 annotated images to the dataset, significantly increasing its size.

2) Dataset Enhancement: Data Augmentation: To expand the scale bar dataset, we used a simple image processing technique. This involved identifying the scale bar by thresholding the pixel values and detecting the darkest horizontal object as the scale bar. Once identified, we generated augmented versions of each image, including variations with brown and white scale bars.

B. Models

We divided the task into two components: a landmark detection component and an optical character recognition (OCR) problem.

*1) **You Only Look Once (YOLO)**:* Introduced in 2015, YOLO [7] is a state-of-the-art, real-time object detection algorithm. It models the object detection problem as a regression task, using a single convolutional neural network (CNN) to spatially predict bounding boxes and assign probabilities to each detected object. Since then many improvements were introduced to the algorithm. At its latest iteration YOLOv11, it can support multiple computer vision tasks, such as detection, segmentation, classification, and pose estimation. This is done by adapting different heads for each task [5].

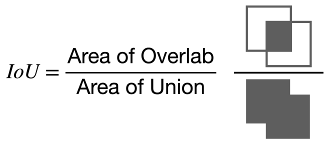
*2) **EasyOCR**:* is a ready-to-use OCR with 80+ supported languages [8]. The system uses CNNs for image feature extraction and a Long-Short-Term-Memory (LSTM) to interpret the extracted features' sequential context and finally, a decoder component to return the detected text.

*3) **DeepLSD**:* [9], a novel line detection method leveraging a Convolutional Neural Network (CNN). This approach generates a "line attraction field," highlighting the likely positions and orientations of line segments within an image.

C. Evaluation Metrics

1) Standard Evaluation: In addition to object classification accuracy, the bounding box detection metrics are:

- **Average Precision (AP)**, is the area under the precision-recall curve (AUC), it summarizes the trade-off between precision and recall at different confidence thresholds.



- **mAP@[0.5:0.95]**: Mean average precision calculated at an intersection over union (IoU) thresholds from 0.5 to 0.95 in steps of 0.05.

2) Morphology Specific Evaluation: To help quantify the accuracy of the model, we used the following metrics:

- **Weber Length Error** The Weber's length (*WL*) is calculated as the Euclidean distance between the landmarks. The error is given by the difference between predicted and ground truth lengths normalized by the true Weber length.

$$Error = \frac{|WL_{GT} - WL_{Pred}|}{WL_{GT}}$$

where WL_{GT} is the ground truth Weber's length, and WL_{Pred} is the predicted Weber's length.

- **AN**: The average Euclidean distance between each predicted landmark and the corresponding ground truth landmark

$$AN = \sqrt{(x_{pred} - x_{GT})^2 + (y_{pred} - y_{GT})^2}$$

- **AM**: The average Euclidean distances normalized by the Weber length (to account for variations in scale)

$$AM = \frac{AN}{WL_{GT}}$$

III. Experimental Results

To address the task, we divided it into two subproblems: **landmark detection** and **optical character recognition (OCR)**.

A. Landmark Detection

For the landmark detection problem, we approached it in three steps:

1) Predicting key points for the thorax only.
2) Detecting key points for the scale bar only.
3) Simultaneously detecting key points for both.

Starting with thorax only prediction, we experimented with two variants of the YOLOv11 models from the Ultralytics Library [5]:

- **YOLO-s**: A smaller model with 9.4M parameters and 21.5B FLOPs.
- **YOLO-l**: A larger model with 25.3M parameters and 86.9B FLOPs.

Both models were fine-tuned on a single Tesla V100 GPU.

Training Schemes:

We evaluated the performance using four fine-tuning schemes over 100 epochs:

1) **Vanilla**:
   - Uses Ultralytics default hyperparameter values for the optimizer.
   - Uses **Mosaic augmentation**, a technique that combines four images into one during training. This augmentation is applied until the final 10

epochs of training to improve the model's ability to generalize across object sizes, aspect ratios, and contexts.

2) **Early Stopping**:
   - Implements a patience of 10 epochs, halting training if no improvement is observed on the validation set for 10 consecutive epochs.
3) **Additional Data Augmentations**:
   - Incorporates various augmentations, including Rotation, Translation, Scaling, Shearing, Perspective, Hue adjustment, HSV-value augmentation, Vertical flipping, and Horizontal flipping (refer to table III for details).
4) **Using both early stopping and additional data augmentations**

The results of these experiments are summarized in Table I, here we only used the 200 annotated images.

		mAP	Error	AN-thorax-start	AM-thorax-start	AN-thorax-end	AM-thorax-end
YOLO-s	Vanilla	0.602	0.0824	20.799	0.1178	24.776	0.0994
	w/ early_stop	0.237	0.4969	49.713	0.344	80.425	0.212
	w/ augmentations	0.481	0.0439	17.004	0.081	16.543	0.0825
	w/ early_stop + aug.	0.155	0.3924	68.511	0.232	54.436	0.297
YOLO-l	Vanilla	0.633	0.0527	12.962	0.0742	17.967	0.0523
	early_stop	0.343	0.5135	67.0321	0.2849	63.911	0.3011
	w/ augmentations	0.453	0.0658	12.359	0.0892	18.956	0.0613
	w/ early_stop + aug.	0.183	1.107	123.997	0.566	124.839	0.5587

Table I

To improve the thorax detection, we employed pseudo labeling and fine-tuned both the YOLO-s and YOLO-l in the vanilla configuration and with augmentations. the results are reported in table II The following figure presents a comparison of the predictions from the two vanilla models, clearly demonstrating accurate landmark predictions.

		mAP	Error	AN-thorax-start	AM-thorax-start	AN-thorax-end	AM-thorax-end
YOLO-s	Vanilla	0.650	0.0359	8.536	0.0479	9.375	0.0446
	w/ augmentations	0.621	0.0384	8.368	0.0450	9.133	0.0422
YOLO-l	Vanilla	0.636	0.0355	7.454	0.0415	8.3525	0.0379
	w/ augmentations	0.577	0.0347	8.049	0.0443	8.9555	0.0406

Table II

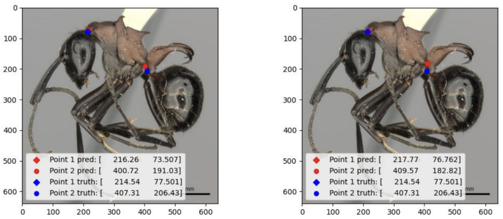


Figure 2. Right side showing the YOLO-s detection and Left side showing the YOLO-l predictions

For the scale bar keypoint detection, we tried using the YOLO models, however, the models were unable to detect the scale-bar object. Instead we opted for two alternatives: a heuristic image processing approach and off-the-shelf deep learning based model.

For the naive approach, we assume that the scale bar is always a black, straight line, located at the bottom between pixel rows 550 and 640, and use the color pixel value to isolate the darkest (black) regions.

We also employed the DeepLSD library and extracted the identified line segments, we then identified line segments with angles approaching horizontal and designated the largest segment as the scale bar, as shown in figure 3

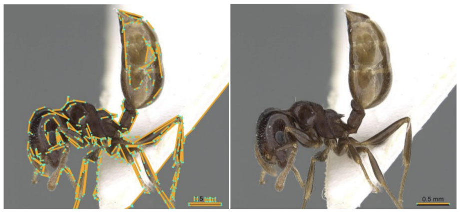


Figure 3. Right side showing the DeepLSD extracted lines and the left side showing the final detected scale bar

B. Optical Character Recognition

Out of the box, EasyOCR successfully detected the correct text bar when evaluated on the set of 200 annotated images. The bounding boxes for the characters forming the scale bar label were correctly identified in 100% of cases. The characters themselves were accurately recognized 96.8% of the time.

Table III
HYPERPARAMETER VALUES

Hyperparameter	Value
init_lr	0.01
final_lr	0.2
momentum	0.937
weight_decay	0.0005
warmup_epochs	3.0
warmup_momentum	0.8
warmup_bias_lr	0.1
degrees	10.0
translate	0.1
scale	0.5
shear	2.0
perspective	0.0
hsv_h	0.015
hsv_s	0.7
hsv_v	0.4
fliplr	0.5
mosaic	1.0
mixup	0.2

IV. DISCUSSION

A. Thorax Detection

From the results in Table I, we observe that the larger models generally performed worse than their smaller counterparts. This suggests that larger datasets are necessary for fine-tuning larger models. Interestingly, early stopping appeared to benefit the object detection task (with smaller mAP values), but it almost always worsened the keypoint detection task (resulting in larger morphological errors). Additionally, in both the vanilla and early stopping configurations, incorporating data augmentations significantly improved performance. It is also important to note that when the size of the training dataset is increased, both models perform equally well. As a result, we can completely opt for the smaller model, as demonstrated by table II and figure 2

B. Scale-bar Detection

We attempted to augment the scale bar dataset by using the naive image processing approach described in II-A2 and train YOLO again but it also failed to detect the scale bar. However, since both DeepLSD and EasyOCR work well out of the box, they can be used to solve this task.

C. Computational Complexity

A notable challenge of our proposed framework lies in its relatively high inference cost. Successfully annotating an image requires utilizing multiple components, including YOLO, DeepLSD, and EasyOCR. This underscores the importance of further optimizing the framework to enhance its efficiency and reduce computational demands.

V. Conclusion

This project highlights the effectiveness of deep learning in automating ant morphological measurements, with a focus on Weber's length estimation. By evaluating various model architectures and training strategies, we demonstrated that a compact YOLO model (9.4M parameters) can deliver remarkable performance when trained on sufficiently large datasets, achieving a mean Average Precision (mAP) of 0.650 and a low normalized Weber length error of 0.0359.

Our use of pseudo-labeling to expand the dataset from 200 to 1850 images underscores the scalability of this approach for entomological research. This strategy minimizes the need for labor-intensive manual annotations while preserving accuracy. Furthermore, we showed that targeted data augmentation techniques significantly enhance model performance, particularly when working with limited datasets.

We recommend the following as future directions

- **Efficiency:** Optimizing computational performance to reduce inference time and resource usage.
- **Feature Expansion:** Extending detection capabilities to additional morphological traits.
- **Accessibility:** Creating a user-friendly interface for broader adoption.
- **Generalization:** Exploring cross-species applicability of the model.

VI. Ethical Risk Assessment

To the best of our knowledge we have not identified any ethical risks. In evaluating the ethical dimensions of our project, we began by considering the possible stakeholders:

- Researchers and Entomologists: The primary users of the system, relying on it for morphological measurements. Their role involves analyzing ant specimens to advance biological research.
- Global Research Community: Broader academic and scientific circles who may adopt or critique the methodology.
- Environment: Machine learning models often involve the use of large servers and GPUs which can harm the environment. More specifically, as our work involves ecological data, there is an indirect connection to biodiversity preservation and potential ecological ramifications.

To ensure the absence of ethical risks, we undertook the following steps:

- Data Transparency: All datasets used for training and evaluation were sourced from publicly accessible repositories with appropriate permissions. Annotations were carefully reviewed to the best of our abilities and the processes clearly mentioned and documented to ensure accuracy and alignment with research goals.
- Model Accuracy and Bias Assessment: Rigorous evaluation metrics ensured that our models were fair and unbiased. We created randomly sampled train/ val/ test splits to ensure that the system generalizes effectively without favoring specific conditions or species
- Environmental Considerations:
  - Impact on Biodiversity Research: our project supports biodiversity preservation by improving the efficiency and accuracy of ant morphological measurements, potentially aiding conservation efforts
  - Reduce Digital footprint: Computational efficiency was prioritized and shown to be feasible and desired. To minimize the environmental footprint of training and running our deep learning models, all checkpoints were shared to ensure that no unnecessary computation is done in the future.

Through this thorough assessment, we concluded that no significant ethical risks exist for our project.

References

[1] T. T. Høye, J. Ärje, K. Bjerge, O. L. Hansen, A. Iosifidis, F. Leese, H. M. Mann, K. Meissner, C. Melvad, and J. Raitoharju, "Deep learning and computer vision will transform entomology," *Proceedings of the National Academy of Sciences*, vol. 118, no. 2, p. e2002545117, 2021.

[2] M. R. Pie and J. Traniello, "Morphological evolution in a hyperdiverse clade: the ant genus pheidole," *Journal of Zoology*, vol. 271, no. 1, pp. 99–109, 2007.

[3] C. E. Sosiak and P. Barden, “Multidimensional trait morphology predicts ecology across ant lineages,” *Functional Ecology*, vol. 35, no. 1, pp. 139–152, 2021.

[4] A. Mathis, P. Mamidanna, K. M. Cury, T. Abe, V. N. Murthy, M. W. Mathis, and M. Bethge, “Deeplabcut: markerless pose estimation of user-defined body parts with deep learning,” *Nature neuroscience*, vol. 21, no. 9, pp. 1281–1289, 2018.

[5] G. Jocher and J. Qiu, “Ultralytics yolo11,” 2024. [Online]. Available: https://github.com/ultralytics/ultralytics

[6] W. M. Teles, L. Weigang, and C. G. Ralha, “Antweb-the adaptive web server based on the ants’ behavior,” in *Proceedings IEEE/WIC International Conference on Web Intelligence (WI 2003)*. IEEE, 2003, pp. 558–561.

[7] J. Redmon, “You only look once: Unified, real-time object detection,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.

[8] JaidedAI, “Easyocr,” https://github.com/JaidedAI/EasyOCR, n.d., accessed: December2024.

[9] R. Pautrat, D. Barath, V. Larsson, M. R. Oswald, and M. Pollefeys, “Deeplsd: Line segment detection and refinement with deep image gradients,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 17 327–17 336.