YOLO-Pole: A Deep Learning Framework for Precise Pole Localization in Aerial Orthophotos

Xiangyu Zhuo and Jiaojiao Tian, *Senior Member, IEEE*

***Abstract*—Pole detection in aerial orthophotos is a critical yet challenging task due to the small size of poles (often reduced to just 1-2 pixels), limited vertical profile visibility, and varying lighting conditions in aerial imagery. Existing approaches primarily rely on bounding box detection, which lacks the precision needed for practical applications such as urban infrastructure mapping and autonomous navigation. In contrast, this paper introduces YOLO-Pole, a novel end-to-end deep learning framework based on You Only Look Once version 7 (YOLOv7) architecture, specifically designed for high-precision pole localization in aerial orthophotos. Instead of providing a coarse bounding box, YOLO-Pole directly predicts the precise pole footprint using a single-stage process. To further refine localization, we introduce a pointwise loss function based on Euclidean distance. Experimental results on a custom dataset with 20 cm ground sample distance (GSD) demonstrate significant improvements in localization accuracy of poles over the standard YOLO model, confirming that precise pole localization is achievable and offering potential for image-based geolocalization.**

***Index Terms*—Pole detection, YOLO, Aerial orthophoto.**

I. Introduction

Accurate detection of poles, such as traffic lights, streetlights, and sign poles, is essential for urban infrastructure management and autonomous navigation systems [1]. As stable vertical structures, poles serve as ground-control points (GCPs) for georeferencing, improving the positional accuracy of aerial imagery and supporting high-definition (HD) mapping. Their precise localization also aids autonomous systems by providing reliable landmarks for navigation.

While existing research focuses on street-view or unmanned aerial vehicle (UAV) imagery [2], where poles are larger and more distinguishable, detecting poles in aerial images is more challenging due to their small size, which is often reduced to just 1-2 pixels. Conventional pole detection methods rely on hand-crafted features like edge detection or texture analysis [3] and struggle with shadows, occlusions, and background noise, requiring extensive manual intervention.

Recent advances in deep learning, particularly models such as Mask R-CNN [4] and YOLO [5], have achieved notable success in small object detection. These models are primarily applied to street-view and UAV imagery [2], where poles are more visible. However, aerial orthophoto-based pole detection remains challenging due to the small size and complex backgrounds. A limited number of studies have addressed this challenge. For example, [1] uses shadow information to detect poles from aerial imagery and co-register with radar imagery to improve the geolocation accuracy of aerial imagery. But the reliance on shadow visibility limits its applicability under varying illumination conditions.

To address these challenges, we propose YOLO-Pole, an end-to-end deep learning model for direct pole localization from aerial orthophotos. As shown in Fig 1, YOLO-Pole directly predicts pole locations from the input orthophoto imagery. Our model also incorporates a pointwise loss function based on the Euclidean distance between predicted and ground-truth pole locations, improving localization accuracy for these tiny objects. We validate YOLO-Pole on a custom dataset with 20 cm GSD aerial orthophotos. Our results show significant improvements in detection performance, both in terms of Intersection over Union (IoU) and pointwise localization accuracy. By demonstrating that high-precision pole localization is achievable, we aim to bridge the gap in aerial pole detection research and enable improved geospatial applications, such as aligning aerial imagery with high-accuracy radar datasets for enhanced positioning.

The primary contributions of our work are as follows:

- **Proof of concept study**: To the best of our knowledge, this is the first deep learning-based study for directly localizing pole footprints in aerial orthophotos. Unlike conventional detection methods that provide bounding boxes, our model precisely predicts pole positions in a single-stage process.
- **End-to-End Localization**: YOLO-Pole is an end-to-end model that directly outputs pole footprints, eliminating the need for post-processing.
- **Pointwise Distance-Based Loss Function**: We introduce a loss function based on the Euclidean distance of pole footprints, improving geometric accuracy in detection.

II. Methodology

In this section, we describe the architecture of the proposed YOLO-Pole model, as illustrated in Fig. 1.

A. Network Architecture

Backbone We adopt a pre-trained CSPDarknet53 [6] as the backbone for its strong feature extraction capabilities and computational efficiency, which is also a widely-used backbone in YOLO-based models.

Detection Neck Following the backbone, the Feature Pyramid Network (FPN) [7] is used to build a feature pyramid with lateral connections between feature maps. A series of

Received 17 February 2025; revised 15 April 2025; accepted 26 April 2025. (Corresponding author: Jiaojiao Tian.)

The authors are with the Remote Sensing Technology Institute (IMF), German Aerospace Center (DLR), 82234 Weßling, Germany (e-mail: xiangyu.zhuo@dlr.de; jiaojiao.tian@dlr.de).

Digital Object Identifier 10.1109/LGRS.2025.3566546

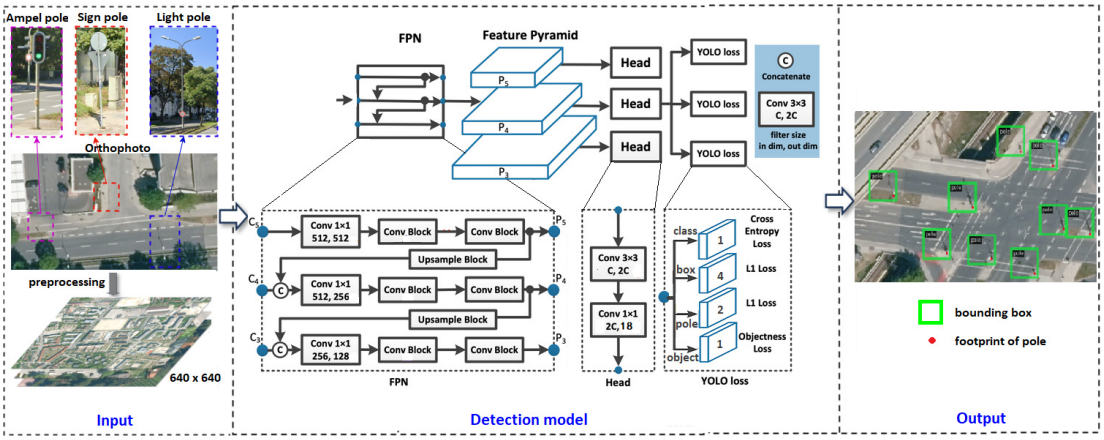


Fig. 1: Technical workflow of the proposed pole detection method.

convolutional layers are employed to refine and concatenate feature maps from different stages of the backbone. Consistent with common YOLO configurations for small object detection tasks, we use the third, fourth, and fifth pyramid levels (P3-P5) as inputs to the detection head because these layers provide a good balance between spatial resolution and semantic abstraction [7], [8], which is critical for detecting small structures like poles in aerial imagery.

Detection Head Our detection head incorporates a dual-branch structure to achieve precise localization of pole footprints. After receiving fused multi-scale features from the neck, the head first applies a 1×1 convolution to reduce feature dimensions, followed by a 3×3 convolution to capture local context. The resulting feature maps are split into two branches: one outputs bounding box parameters (e.g., center, size, objectness), and the other regresses the pole's relative (x, y) location for precise footprint localization.

B. Loss Function

Traditional object detection metrics, such as the IoU, often fall short in scenarios involving extremely small objects, as these metrics tend to lose sensitivity when the objects occupy a minuscule area of the image. This challenge is particularly pronounced in the context of aerial imagery, where objects such as pole footprints may only span 1-2 pixels. The introduction of a pointwise loss, inspired by Xu et al. [9]'s development of the Dot Distance (DotD) metric, addresses this limitation. The DotD metric effectively measures the Euclidean distance between the predicted and actual central points of the objects, providing a direct and highly sensitive indication of localization accuracy.

Therefore, we integrate pointwise loss for poles into our loss function. The total loss function for YOLO-Pole model L including the loss components for bounding box regression, objectness, classification, and pole regression, is defined as:

$$L = \lambda_{\text{bbox}} L_{\text{bbox}} + \lambda_{\text{obj}} L_{\text{obj}} + \lambda_{\text{class}} L_{\text{class}} + \lambda_{\text{pole}} L_{\text{pole}}. \quad (1)$$

- L_{bbox} is the bounding box regression loss calculated using Complete Intersection over Union (CIoU) [10]:

$$L_{\text{bbox}} = 1 - \text{IoU} + \frac{\rho^2(\mathbf{b}, \mathbf{b}_{gt})}{c^2} + \alpha \cdot v, \quad (2)$$

  where $\rho(\mathbf{b}, \mathbf{b}_{gt})$ is the Euclidean distance between the centers of the predicted and ground-truth bounding boxes, c is the diagonal length of the smallest enclosing box covering both bounding boxes, v measures the consistency of aspect ratio, and α is a trade-off parameter.
- L_{obj} is the objectness loss, which measures the model's confidence in predicting the presence of an object:

$$L_{\text{obj}} = \begin{cases} -\log(\hat{y}_{\text{obj}}), & \text{if an object is present,} \\ -\log(1 - \hat{y}_{\text{obj}}), & \text{otherwise.} \end{cases} \quad (3)$$

  where $\hat{y}_{\text{obj}}$ is the predicted probability of an object being present in a given bounding box.
- L_{class} is the classification loss for predicting the correct class of an object using cross-entropy:

$$L_{\text{class}} = -\sum_{c=1}^{C} y_c \log(\hat{y}_c), \quad (4)$$

  where C is the number of classes, y_c is a binary indicator (0 or 1) if class label c is correct for the observation, and $\hat{y}_c$ is the predicted probability of class c.
- L_{pole} is the pole regression loss, which is calculated using the Euclidean distance for a precise localization:

$$L_{\text{pole}} = \sqrt{(p_x - \hat{p}_x)^2 + (p_y - \hat{p}_y)^2}, \quad (5)$$

  where p and $\hat{p}$ denote the actual and predicted pole center coordinates (x, y), respectively.
- $\lambda_{\text{bbox}}, \lambda_{\text{obj}}, \lambda_{\text{class}}, \lambda_{\text{pole}}$ are the weighting factors for each respective loss component. The specific weights used are detailed in Section III.

III. Experiment

A. Dataset

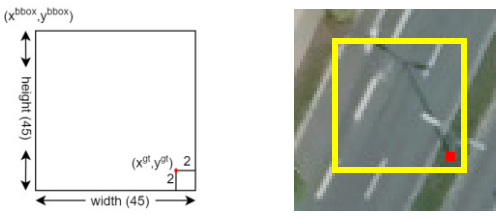
The dataset consists of true orthophotos with 20 cm resolution, chosen as a challenging test case where pole footprints

often span only 1–2 pixels. This setup serves as a strict benchmark for evaluating localization under limited spatial detail. The images are collected from the Munich and Lindau regions in southern Bavaria, covering diverse rural and urban scenes (Fig. 2). The *pole* class includes traffic signals, streetlights, and sign poles, each manually annotated with a precise footprint and bounding box. To ensure high annotation quality, only fully visible poles are labeled. In total, 83 orthophotos (3500 × 3500 pixels) with 7763 annotated poles are used.

(a) (b)

Fig. 2: Orthophoto images used in Experiment. (a) A rural area in Lindau, Germany. (b) An urban area in Munich, Germany.

Object detection methods generally require bounding boxes as input. Therefore, pole annotations are converted to bounding boxes. Since shadows are key texture features, the boxes are designed to encompass pole shadows. However, larger boxes can capture irrelevant objects, introducing noise. To balance inclusiveness and accuracy, we use square bounding boxes of 45×45 pixels. To address manual annotation errors, we position the pole near the bottom right corner of the box, with a 2-pixel vertical and horizontal offset, as shown in Fig. 3, where (x^{bbox}, y^{bbox}) denotes the top left corner of the bounding box and (x^{gt}, y^{gt}) denotes the ground-truth of the pole. This ensures the pole is fully enclosed despite the existence of potential annotation errors. Fig. 4 illustrates typical ground-truth annotations, where yellow rectangles mark bounding boxes, and red dots represent pole footprints. The shadows show varying shapes and azimuth angles due to differing sun positions at the time of acquisition.



(a) (b)

Fig. 3: Bounding box and pole ground-truth. (x^{bbox}, y^{bbox}) in (a) denotes the top left corner of the bounding box and (x^{gt}, y^{gt}) denotes the ground-truth of the pole. In (b), bounding box is depicted in yellow and pole is highlighted in red.

B. Experimental Setup

To account for diverse pole shadows in our dataset, we applied several data augmentation techniques. The original images (3500 × 3500 pixels) were first cropped into 640 × 640 patches. Since YOLO is not inherently rotation-invariant, our preliminary attempts showed detection failure when using only unrotated images, mainly due to differences in shadow directions between training and test data. While applying a full range of rotation angles would significantly increase the training cost, we found that rotating each patch by 20° and 40° (Fig. 5) provides a good trade-off between angular diversity and training efficiency. In addition to these rotations, standard augmentation techniques such as flipping, color jittering, and affine transformations were also used to further enhance generalization.

We use a YOLOv7-tiny backbone with Path Aggregation Feature Pyramid Network (PAFPN) neck, TinyDownSampleBlock, and LeakyReLU activation. The model is trained using the MMDetection framework [11] on an Ubuntu 20.04.6 LTS system with an NVIDIA RTX 3090 GPU. We initialize the model with COCO-pretrained weights [12]. The dataset is split into 50 orthophotos for training/validation and 33 for testing, ensuring no spatial overlap. Input images are resized to 640×640 pixels. We train the model with a batch size of 16 for 200 epochs using SGD optimizer with an initial learning rate of 0.01, momentum of 0.937, Nesterov acceleration, and a weight decay of 0.0005.

We adopt the default YOLO augmentations (e.g., mosaic, flipping, color jittering, affine transformations). The loss weights are manually set to focus on precise point-level localization. Specifically, we use $\lambda_{\text{bbox}} = 0.1$, $\lambda_{\text{obj}} = 1.0$, $\lambda_{\text{class}} = 0.0$, and $\lambda_{\text{pole}} = 2.0$. This configuration reflects the nature of our task, where pole footprints are extremely small, and pointwise accuracy is more critical than bounding box quality. These values were selected through preliminary experiments and manual tuning based on validation performance.

C. Results Visualization

The evaluation is carried out on the test set comprising 33 orthophoto images from Milbertshofen and Lindau. Figure 6 illustrates the visualization of pole detection results in various scenarios, where Ground-truths are marked by red crosses and predictions are marked by yellow crosses. It can be seen that in most scenarios, many poles have been correctly detected and localized. However, as the ground-truth annotation is not perfect, some poles are overlooked during annotation although they are well visible on the orthophoto, as shown in subfigure (d), therefore the number of predicted poles is generally larger than the number of ground-truth annotations.

D. Evaluation Criteria

To evaluate the precise localization of pole footprints, we use both area-based and distance-based metrics. For area-based metrics, the IoU metric compares predicted and ground-truth bounding boxes, but since the bounding boxes (45×45 pixels) are much larger than poles (1–2 pixels), IoU alone is not representative enough. To address this problem, we calculate the Euclidean distance between predicted and ground-truth poles as well, using a threshold of 5 pixels (1 meter) for positional accuracy.

Fig. 4: Ground truth of different types of poles. Poles are annotated as red dots enclosed by yellow bounding boxes: (a) Sign pole, (b) Traffic light pole, (c) Single-arm streetlight pole, (d) Double-arm streetlight pole.

Fig. 5: Shadow simulation via image rotation: (a) Original patch, (b) Patch rotated by $20°$, (c) Patch rotated by $40°$.

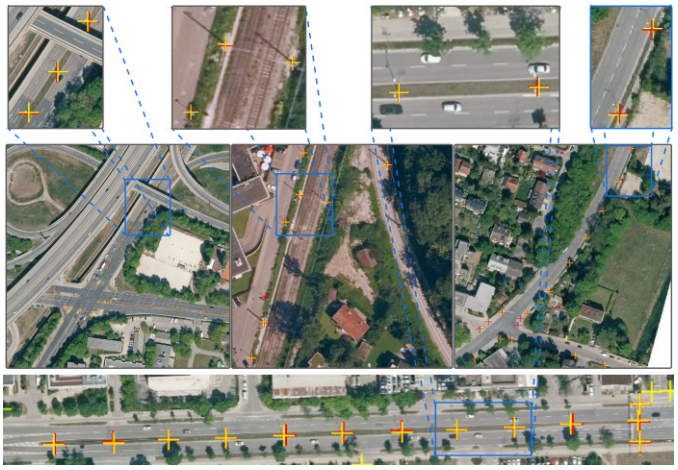


Fig. 6: Detected poles (yellow) and ground-truth (red).

Our evaluation metrics are defined as:

- **True Positive (TP)**: A detected pole is within 5 pixels of a ground-truth pole.
- **False Positive (FP)**: A detected pole exceeds the 5-pixel distance from any ground-truth pole.
- **False Negative (FN)**: A ground-truth pole has no detected pole within 5 pixels.

Fig. 7 shows examples of detection results. The first row illustrates from left to right true positive samples with distances of 1, 3, and 5 pixels, respectively. The second row shows false negatives where the model missed to detect poles, while the third row presents from left to right three false positive cases caused by wrong semantics, distance exceeding the threshold, and incorrect ground-truth labels.

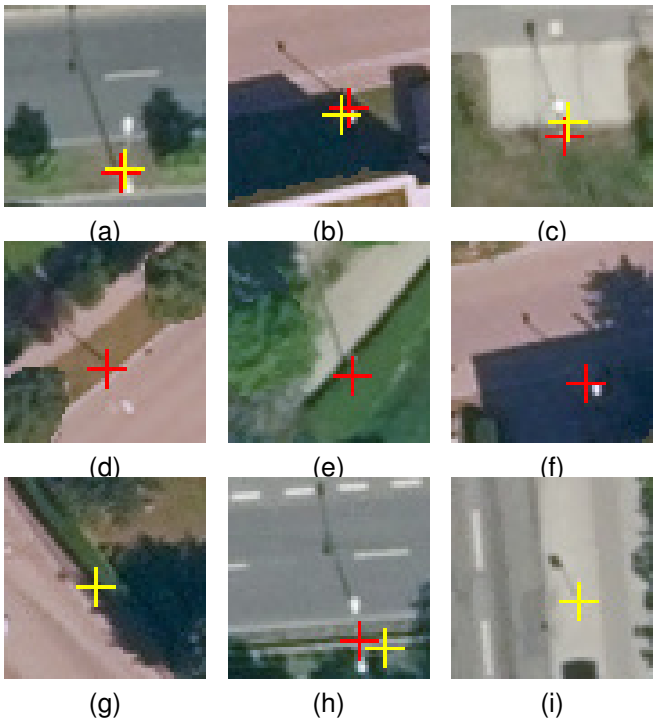


Fig. 7: Examples of detection outcomes. Red: ground truth, yellow: predictions. (a–c) TP; (d–f) FN; (g–i) FP.

E. Quantitative Evaluation

Table I presents a comparative analysis of performance metrics among standard YOLOv7, Faster R-CNN [13], SSD [14], YOLOv8 [15] and our proposed YOLO-Pole. The evaluation includes mean IoU and mean Average Precision (mAP) as area-based metrics, and precision, recall, and F1 score as distance-based metrics. YOLO-Pole achieves notable improvements over the standard model. Notably, the bounding boxes used for evaluation are much larger than the actual poles, thereby incorporating irrelevant background and leading to lower IoU and mAP values than typically seen in object detection tasks. Additionally, we include an ablation variant, YOLO-Pole$_{\text{IoU}}$, which uses conventional IoU-based loss without the pointwise Euclidean loss component. As shown in Table I, YOLO-Pole$_{\text{IoU}}$ performs marginally below Standard YOLOv7, because the pole footprint branch becomes inactive

Fig. 8: False positive samples resulting from misannotations.

without the pointwise Euclidean loss and may introduce additional noise. It can be seen that the pointwise loss is essential for achieving higher localization precision.

It is important to note that the ground-truth data is not exhaustively annotated. As annotators typically label only the poles they are confident are visible, poles obscured by shadows or partially visible are often omitted. Consequently, some semantically correct predictions are not annotated and are marked as false positives, as shown in Fig. 8.

TABLE I: Comparison of performance metrics between baseline methods and YOLO-Pole.

Model	mean IoU	mAP	Precision	Recall	F1 Score
Standard YOLOv7	0.690	0.668	0.421	0.369	0.393
Faster R-CNN	0.712	0.682	0.419	0.390	0.404
SSD	0.645	0.621	0.390	0.330	0.357
YOLOv8	0.715	0.690	0.450	0.351	0.395
YOLO-Pole	**0.752**	**0.711**	**0.485**	**0.413**	**0.446**
YOLO-Pole$_{\text{IoU}}$	0.686	0.665	0.420	0.367	0.391

IV. Discussion and Conclusions

Detecting pole footprints from 20 cm aerial orthophotos is challenging due to the limitations of nadir-view imagery, where poles often appear as small point-like features. Given the scarcity of datasets and relevant methods, this work serves as a feasibility study, demonstrating that high-precision pole localization is achievable under such conditions.

Unlike conventional bounding box-based approaches, YOLO-Pole directly predicts pole footprints in an end-to-end manner, eliminating post-processing steps. The integration of pointwise Euclidean loss further enhances localization accuracy, confirming the applicability of deep learning to this task.

This study is an initial step toward automated pole localization in aerial imagery. We focus on 20 cm orthophotos, as DOP 20 imagery is freely available for many European cities, offering significant potential for large-scale application of this approach. The approach is also expected to perform even better on higher-resolution imagery (e.g., 10 cm), where pole footprints are more clearly visible. In addition, the framework can be extended to other point-like objects, such as utility markers or fire hydrants, which are also difficult to detect using traditional methods. We hope this work encourages further research on expanding datasets and incorporating additional data sources to improve detection performance and generalizability.

References

[1] T. Krauß, F. Kurz, and H. Runge, "Automatic pole detection in aerial and satellite imagery for precise image registration with sar ground control points," *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 1, pp. 85–91, 2022.

[2] M. M. Alam, Z. Zhu, B. Eren Tokgoz, J. Zhang, and S. Hwang, "Automatic assessment and prediction of the resilience of utility poles using unmanned aerial vehicles and computer vision techniques," *International Journal of Disaster Risk Science*, vol. 11, pp. 119–132, 2020.

[3] B. Cetin, M. Bikdash, and M. McInerney, "Automated electric utility pole detection from aerial images," in *IEEE Southeastcon 2009*, 2009, pp. 44–49.

[4] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[5] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[6] C. Wang, H. M. Liao, I. Yeh, Y. Wu, P. Chen, and J. Hsieh, "Cspnet: A new backbone that can enhance learning capability of CNN," *CoRR*, vol. abs/1911.11929, 2019.

[7] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2117–2125.

[8] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 7464–7475.

[9] C. Xu, J. Wang, W. Yang, and L. Yu, "Dot distance for tiny object detection in aerial images," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 1192–1201.

[10] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, and D. Ren, "Distance-iou loss: Faster and better learning for bounding box regression," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, no. 07, 2020, pp. 12 993–13 000.

[11] K.Chen and et al., "Mmdetection: Open mmlab detection toolbox and benchmark," 2019.

[12] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Computer Vision – ECCV 2014*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham: Springer International Publishing, 2014, pp. 740–755.

[13] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 6, pp. 1137–1149, 2016.

[14] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*. Springer, 2016, pp. 21–37.

[15] G. Jocher and the Ultralytics Team, "Ultralytics yolov8," https://github.com/ultralytics/ultralytics, Ultralytics, London, United Kingdom, 2023, accessed: 2025-05-18.