Military Asset Detection and Classification Using YOLOv8

Cordell Stonecipher
Department of Industrial & Systems Engineering
Oakland University, Rochester, MI, USA
Email: stonecipher@oakland.edu

***Abstract*—Modern defense analytics increasingly relies on automated systems capable of identifying objects of interest in real time. Whether used in reconnaissance, surveillance, or operational planning, computer vision systems allow military analysts to process large volumes of imagery far beyond human capacity. This project investigates the performance of YOLOv8, a state-of-the-art object detector, on a complex dataset containing 12 classes of military and civilian assets. The workflow includes data preprocessing, annotation cleaning, exploratory data analysis (EDA), bounding-box statistical modeling, training of YOLOv8n, and quantitative evaluation using precision, recall, and mean Average Precision (mAP). The study highlights fundamental challenges in machine learning applications involving highly imbalanced datasets and small-object detection. Results reveal strong performance on visually prominent and heavily represented classes such as tanks and aircraft, while extremely poor detection accuracy is observed for underrepresented categories like civilians and trenches. The findings offer insights into model behavior, data structure, and architectural performance characteristics, demonstrating the importance of thoughtful dataset design, mathematical understanding of loss functions, and detailed analytics in machine learning deployments.**

I. Introduction

In military and defense settings, rapid and accurate identification of battlefield elements is crucial. From aerial reconnaissance to border monitoring, automated object detection systems significantly enhance situational awareness by providing consistent and scalable recognition capabilities. With the proliferation of high-resolution sensors and unmanned aerial vehicles (UAVs), the volume of imagery requiring analysis has grown substantially, motivating the need for machine learning models that can detect and classify assets reliably and efficiently.

Traditional computer vision techniques rely on manually engineered features such as SIFT or HOG descriptors, which typically perform poorly in real-world defense imagery due to environmental variability, inconsistent lighting conditions, and occlusion. Deep learning, particularly convolutional neural networks (CNNs), has transformed this area by enabling models to learn multi-scale hierarchical features directly from data.

YOLO (You Only Look Once) revolutionized object detection by proposing a single-stage approach [1], offering real-time inference while maintaining competitive accuracy. YOLOv8, the latest iteration from Ultralytics [2], incorporates several architectural refinements such as an anchor-free design, enhanced data augmentation pipeline, and decoupled classification and regression heads. These improvements make it a compelling candidate for investigating object detection in complex defense environments.

This project integrates data science, mathematical modeling, and real-world computer vision techniques to evaluate YOLOv8 on a challenging military dataset. The emphasis is not only on training a model but also on understanding how dataset characteristics shape learning behavior.

II. Problem and Data Description

The central problem is to detect and classify military and civilian assets across a diverse image set. The dataset consists of 26,315 labeled images containing 50,822 objects belonging to 12 classes. The variation in geography, lighting, perspective, and environmental conditions introduces substantial real-world complexity.

A. Dataset Structure and Characteristics

The dataset includes the following categories:

- Personnel (soldier, camouflage_soldier, civilian)
- Vehicles (military_tank, military_truck, military_vehicle, civilian_vehicle)
- Heavy equipment (military_warship, military_aircraft, military_artillery)
- Others (weapon, trench)

Each annotation file follows the standard YOLO format:

$$(class\_id,\ x_c,\ y_c,\ w,\ h)$$

with all coordinates normalized to $[0,1]$. This normalization makes the labels invariant to image resolution and simplifies preprocessing.

B. Example Images

Figures 1 and 2 illustrate representative images.

These samples highlight variability in object sizes, occlusions, and terrain complexity.

C. Parsing and Cleaning Labels

Numerous annotation files contained malformed tokens such as ”0.23civilian” or stray characters. To address this, a robust parser was implemented:

Fig. 1. Example labeled training image containing tanks and military vehicles.

Fig. 2. Complex urban scene with multiple tanks detected.

```
def safe_float(token: str):
    cleaned = ''.join(ch for ch in token
                      if ch.isdigit() or ch in '.-')
    if cleaned in {'', '.', '-'}:
        raise ValueError("Malformed numeric value")
    return float(cleaned)
```

This function ensures safe extraction of numeric values, maintaining data integrity for training.

III. Exploratory Data Analysis

Exploratory Data Analysis (EDA) plays a crucial role in understanding the dataset's structure, distributional properties, and potential challenges.

A. Class Distribution

Figure 3 shows severe class imbalance, with tanks dominating the dataset. This imbalance directly affects gradient flow during training, causing rare classes to contribute disproportionately less to parameter updates.

B. Bounding Box Scale Distribution

Bounding-box area, defined as:

$$A = w \times h,$$

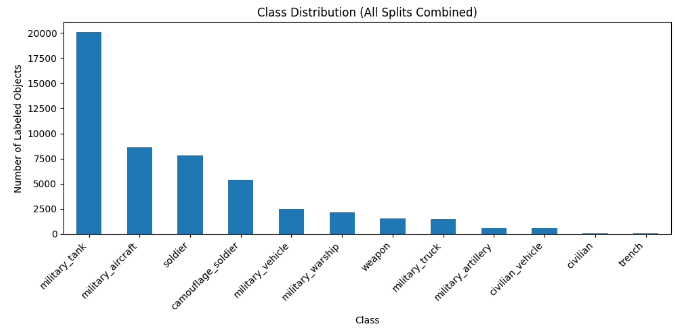

Fig. 3. Class distribution across all labeled objects.

provides a measure of object scale. The distribution shown in Figure 4 reveals that the majority of objects occupy less than 5% of the image, explaining why YOLO struggles to detect many of them.

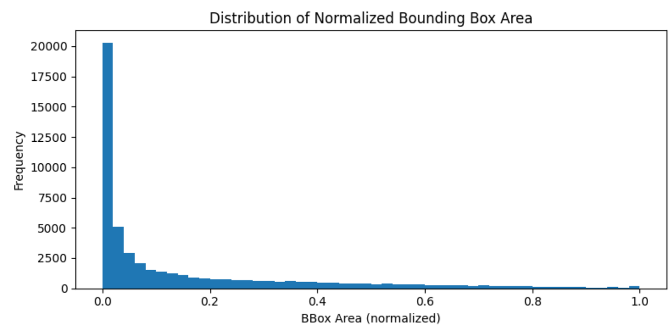

Fig. 4. Histogram of normalized bounding-box areas.

C. Class-Wise Average Bounding Box Sizes

Figure 5 highlights substantial variations in object scale across classes.

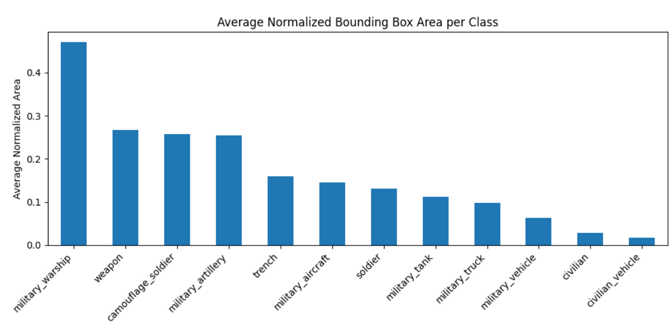

Fig. 5. Average bounding-box area per class, revealing large inter-class differences.

Warships, artillery, and aircraft tend to appear large, while civilians and civilian vehicles appear very small.

IV. Methodology

A. YOLOv8 Architecture

YOLOv8 predicts bounding boxes and classifications in a single forward pass. Unlike earlier YOLO versions that used anchor-based detection, YOLOv8 adopts an anchor-free design, representing bounding boxes explicitly via center coordinates and width-height parameters.

The output tensor includes:

$$(x,\ y,\ w,\ h,\ P_{obj}, P(c_1), ..., P(c_K)),$$

where P_{obj} is objectness confidence and $P(c)$ are class probabilities.

B. CIoU Loss and Bounding Box Optimization

Bounding box regression uses Complete IoU Loss [4]:

$$CIoU = IoU - \frac{\rho^2(b, b_{gt})}{c^2} - \alpha v.$$

This formulation balances:

- overlap accuracy,
- center point distance,
- aspect ratio consistency.

C. Training Configuration

The model was trained using:

- Learning rate schedule with warmup
- Image size: 640×640
- Batch size: 32
- Augmentations: mosaic, scaling, color jitter

D. Evaluation Metrics

Evaluation followed COCO mAP conventions [5], computing:

$$Precision = \frac{TP}{TP + FP}, \quad Recall = \frac{TP}{TP + FN},$$

and:

$$mAP = \int_0^1 P(R)dR.$$

mAP@50–95 averages IoU thresholds from 0.50 to 0.95, providing a stringent measure of localization.

V. Results

A. Overall Model Performance

The YOLOv8n model achieved:

- Precision: 0.715
- Recall: 0.507
- mAP@50: 0.560
- mAP@50–95: 0.375

The decline from mAP@50 to mAP@50–95 indicates difficulty in achieving tight bounding boxes.

B. Per-Class Performance

High-frequency classes excelled:

$$mAP_{aircraft} = 0.897, \quad mAP_{tank} = 0.873.$$

Rare classes failed entirely:

$$mAP_{civilian} = 0, \quad mAP_{trench} = 0.$$

VI. Discussion

A. Impact of Class Imbalance

Rare classes contributed almost no meaningful gradient updates, resulting in poor learned representations.

B. Small Object Detection Limitations

CNN downsampling reduces spatial resolution, disproportionately harming detection of small objects.

C. Model Interpretation

The model's high precision but moderate recall suggests conservative prediction behavior, likely due to confidence thresholding.

VII. Conclusion

This project demonstrated the development and evaluation of a complete data analytics pipeline for military asset detection using the YOLOv8 deep learning architecture. Through a combination of rigorous data preprocessing, exploratory analysis, model training, and quantitative assessment, several significant insights emerged regarding both the strengths and limitations of modern computer vision systems in defense-oriented contexts.

First, the results clearly show that YOLOv8 is highly effective at detecting large, visually distinctive, and frequently represented classes within the dataset. The exceptionally high mAP scores for categories such as *military_tank* and *military_aircraft* demonstrate the architecture's ability to learn consistent feature representations even under varying environmental conditions. These findings align with prior research indicating that YOLO models excel when object scale is sufficient and intra-class variation is relatively constrained. For defense applications, this suggests that YOLOv8 can reliably identify key high-value targets that typically appear prominently in reconnaissance imagery.

However, the project also exposed persistent challenges in small-object detection and long-tailed class distribution learning. Classes such as *civilian* and *trench* suffered from near-zero detection performance, a direct consequence of their very small bounding boxes and extreme scarcity within the dataset. From the perspective of convolutional neural network architectures, this outcome is expected: deep model layers progressively downsample spatial features, causing small objects to lose representational fidelity. The imbalance in label frequency further exacerbates this, as gradient updates are dominated by the most frequent classes. This phenomenon mirrors the "long-tail problem" extensively documented in the literature, where rare categories fail to contribute meaningfully during optimization. In real-world defense settings, this presents a risk: systems trained on imbalanced datasets may fail to detect subtle but operationally significant objects, such as civilians or camouflaged structures.

The exploratory data analysis conducted in this project underscored how critical it is to understand the statistical structure of the input dataset before model training. Visualizations revealed not only class imbalance and bounding-box skewness but also multi-scale variability across object types. These dataset attributes directly influenced performance outcomes and provided a framework for interpreting model behavior. In this way, EDA proved indispensable, reinforcing a core principle of data analytics: model performance is inseparably tied to the quality, distribution, and geometry of the input data.

Another important finding involved the difference between mAP@50 and mAP@50–95 scores. The substantial drop between these metrics highlights YOLOv8's difficulty in performing precise localization, even when it can roughly identify the correct object. This behavior is characteristic of anchor-free models, which optimize bounding-box regression in a unified representation rather than through predefined anchor priors. For defense applications where precise geospatial localization is required, such as automated targeting or object tracking, this limitation underscores the need for further refinement.

The project also demonstrated the importance of robust preprocessing. The dataset contained malformed annotation files and irregular label formats, requiring custom numerical parsing and verification procedures. Such inconsistencies, if left unaddressed, can lead to silent model failures or corrupted training batches. This reflects a broader truth in applied machine learning: a substantial portion of model performance hinges not on the architecture itself, but on the quality and structure of the data pipeline.

Looking ahead, several directions offer promising avenues for improving performance. Techniques such as class-balanced sampling, focal loss, synthetic minority oversampling, and data augmentation targeted at rare categories could help mitigate the long-tail issue. Multi-scale feature-enhancement modules or larger YOLOv8 variants may improve small-object detection. Additionally, incorporating temporal information from video-based inputs, rather than relying exclusively on static images, could allow models to exploit motion cues to improve robustness in operational scenarios.

In summary, this project successfully applied YOLOv8 within a modern data analytics framework to address a meaningful problem in defense-oriented object detection. The work highlights both the impressive capabilities of contemporary deep learning models and the persistent challenges associated with imbalanced datasets, small-object detection, and imperfect annotation quality. The insights gained reinforce the importance of comprehensive exploratory analysis, rigorous preprocessing, mathematically grounded architectural understanding, and thoughtful evaluation when deploying machine learning systems in mission-critical environments. Ultimately, this study underscores that effective object detection is not solely a function of model choice but is fundamentally shaped by the interplay between data, architecture, optimization, and analytical methodology.

APPENDIX: CODE EXCERPTS

A. Label Parsing

```
cls_id = int(parts[0])
x_c = safe_float(parts[1])
y_c = safe_float(parts[2])
w   = safe_float(parts[3])
h   = safe_float(parts[4])
```

B. Data Assembly

```
records.append({
    "split": split,
    "image_path": img_path,
    "class_id": cls_id,
    "class_name": cls_name,
    "x_center": x_c,
    "y_center": y_c,
    "width": w,
    "height": h,
})
```

REFERENCES

[1] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," *IEEE CVPR*, 2016. Available: https://arxiv.org/abs/1506.02640

[2] G. Jocher, "YOLOv8 Documentation," Ultralytics, 2023. Available: https://docs.ultralytics.com

[3] R. Madhuwala, "Military Object Detection Dataset," Kaggle, 2023. Available: https://www.kaggle.com/datasets

[4] Z. Zheng et al., "Distance-IoU Loss for Bounding Box Regression," *AAAI*, 2020. Available: https://arxiv.org/abs/2005.03572

[5] T.-Y. Lin et al., "Microsoft COCO: Common Objects in Context," *ECCV*, 2014. Available: https://arxiv.org/abs/1405.0312

[6] T.-Y. Lin et al., "Focal Loss for Dense Object Detection," *ICCV*, 2017. Available: https://arxiv.org/abs/1708.02002

[7] B. Kang et al., "Decoupling Representation and Classifier for Long-Tailed Recognition," *ICLR*, 2019. Available: https://arxiv.org/abs/1910.09217