



Hyperparameter Optimization and Validation of a Deep Learning Instance Segmentation Model for Real-Time Specific Military Equipment Surveillance

Marko M. Živanović^{1,*} , Bozidar Jankovic² , Ana Savic¹ ,

Nemanja Maček¹ , Hana Stefanović¹

¹ School of Electrical and Computer Engineering, Academy of Technical and Art Applied Studies (VIŠER), Belgrade, Serbia

² Service for Combating High-Tech Crime, Directorate for Technology, Police Directorate, Ministry of Interior, Republic of Serbia

Original scientific paper

computer sciences,
telecommunications, IT

Keywords:

instance segmentation,
specific military equipment,
combat readiness,
hyperparameter optimization,
mAP, F1-Score, machine
learning, real-time
surveillance, camouflage,
YOLO12-seg.

Abstract

Introduction/Objective: On the modern battlefield, a soldier's combat readiness is directly dependent on the proper use of specific military equipment (SME). Automatic verification of SME usage, primarily ballistic helmets and combat vests, is a critical factor for risk reduction and increasing combat effectiveness. Traditional object detection (bounding box) is inadequate for military conditions due to camouflage and occlusion. The objective of this paper is the development, rigorous optimization, and validation of an efficient system for real-time SME segmentation, based on a modern, lightweight YOLO12-seg architecture. **Methods:** The system was trained on a custom-created "Mil-SME-Seg" dataset focused on military personnel. An advanced set of data augmentations (Mosaic, MixUp, HSV-shift) was applied to achieve robustness. A systematic optimization of the key "number of epochs" hyperparameter was performed through five independent, repeatable training runs (100, 120, 150, 200, and 250 epochs) to identify the optimal balance between accuracy and overfitting. Performance was evaluated using a comprehensive set of segmentation metrics, including mAP50–95(M), mAP50(M), mAP75(M), and F1–Score(M). **Results:** Experiments showed that training for 150 epochs yields the best generalized results on the validation set. Peak performance was achieved at epoch 144, with an mAP50–95(M) of 0.6128, mAP50(M) of 0.7852, and an F1–Score(M) of 0.741. Metric analysis indicates that longer training runs (200 and 250 epochs) lead to clear performance degradation due to overfitting. Qualitative analysis confirms the superiority of segmentation in tactically relevant scenarios (camouflage, partial occlusion). **Conclusion:** The optimal number of epochs for training the model on this dataset is approximately 150. The resulting model, integrated into a system with support for dynamic "military scenarios," proved its practical applicability and tactical value, achieving real-time processing speeds (48-55 FPS). This paper lays the foundation for future systems that will use segmentation masks for equipment damage assessment and sensor fusion.

1. Introduction

The modern battlefield is characterized by high dynamics, asymmetric threats, and the fusion of vast amounts of sensor data. In such an environment, the individual soldier is viewed as an integrated system, whose combat effectiveness and survivability directly depend on the serviceability and proper use of specific military equipment (SME). Verifying combat readiness, which includes the wearing of ballistic helmets, combat vests with plates, and other equipment, traditionally relies on visual inspection by commanders, which is inefficient, subjective, and impossible to conduct in real-time during operations.

The development of artificial intelligence (AI) systems, particularly in the domain of computer vision, offers a solution to automate this process. However, the majority of existing research in protective equipment monitoring focuses on industrial safety (e.g., construction helmets) and uses object detection methods [1].

These methods, which produce a rectangular bounding box around the detected object, are fundamentally inadequate for the complexity of the military environment for several reasons [2].

* Corresponding author

E-mail address: markoz@viser.edu.rs; (Marko M. Živanović)

Received: 10 November 2026;

Received in revised form: 17 June 2026;

Accepted: 31 August 2026;

Available online: 1 September 2026;

<https://doi.org/10.5937/vojtechg75-62704>



Military equipment is designed to blend in with the environment (camouflage patterns) [3]. A bounding box, by definition, also includes a significant portion of the background, which creates "noise" in the training data. The model becomes confused, leading to false negatives (missed detections) or false positives. Under conditions of partial occlusion (e.g., a soldier in cover), the box cannot determine which part of the equipment is visible [4].

In military applications, it is not enough to say "helmet." It is crucial to distinguish between types (e.g., a ballistic helmet vs. a standard one) and potentially between friendly and enemy assets. A bounding box provides no information about the object's shape [5].

Bounding box detection provides a binary answer (present/absent). It cannot provide a basis for answering the critical question: "Is the equipment damaged?" (e.g., is there a visible ballistic impact on the vest).

When creating a dataset, annotating a camouflaged object with a bounding box is inherently imprecise. Two different annotators will produce two different boxes. This label noise hinders the model's learning, especially on small, custom datasets.

The solution to these problems lies in instance segmentation. Instead of a coarse box, this method outputs a precise, pixel-level mask that outlines the exact shape of each detected object. The mask solves all the aforementioned problems: (a) it ignores the camouflaged background, (b) it provides an unambiguous signal for learning, (c) it defines the exact shape for identification, and (d) it provides a basis for further analysis of the object's surface [6].

The YOLO (You Only Look Once) architecture has established itself as a standard for real-time processing. Its latest iteration [7] introduces highly efficient models for segmentation. In this paper, we focus on a lightweight, segmentation-oriented 'nano' architecture from the YOLO12-seg, as its low computational requirement (small parameter count, low GFLOPS) is a key prerequisite for in-field deployment, either through processing RTSP streams or via integration onto edge-computing devices [8].

However, the performance of deep learning models critically depends on the training process and the choice of hyperparameters. One of the most important is the number of training epochs. Too few epochs lead to underfitting, while too many lead to overfitting [9], where the model "memorizes" the training data and loses its ability to generalize to new data.

The contribution of this paper is threefold: The creation and detailed description of a custom dataset "Mil-SME-Seg" for segmenting specific military equipment, including the augmentation methodology. Conducting a rigorous experimental analysis to determine the optimal number of training epochs for the model on this dataset, using a comprehensive set of metrics. The demonstration and in-depth discussion of the optimized model's tactical applicability through a prototype system with "military scenarios," proving the superiority of segmentation over detection.

2. Methodology

This chapter details the materials and methods used, including the model's architectural basis, the mathematical apparatus of loss functions and metrics, and the precise design of the experiment conducted. The model used is a modern, lightweight (nano) architecture for instance segmentation from the YOLO12-seg family [10]. YOLO12 introduces an attention-centric design that replaces traditional convolutional feature extraction in the backbone stages with area attention modules, enabling more efficient long-range dependency modeling while maintaining low computational overhead. The architecture comprises three principal components: (1) a backbone based on C3k2 blocks augmented with area attention, which extracts multi-scale feature representations from the input image at four pyramid levels (P2–P5); (2) a neck employing a bidirectional Path Aggregation Network (BiFPN) for multi-scale feature fusion with learnable weights; and (3) a decoupled head that simultaneously produces bounding box predictions, class probabilities, and prototype mask coefficients. The segmentation head generates $k=32$ prototype masks in a dedicated branch, which are linearly combined per instance using predicted coefficients to produce the final pixel-level masks. The "nano" variant used in this work has approximately 2.9 million parameters and requires around 10.5 GFLOPs for a 640×640 input, making it suitable for both real-time GPU inference and potential edge deployment.

The core convolutional neural network, which can be defined as a function B . This function maps an input image $I \in \mathbb{R}^{H \times W \times 3}$ to a hierarchical set of feature maps at different scales:

$$C_l \in \mathbb{R}^{H_l \times W_l \times D_l} \quad (1)$$

where $C_l \in \mathbb{R}^{H_l \times W_l \times D_l}$ represents the feature map at level l . Typically, $H_l < H_{l-1}$ and $W_l < W_{l-1}$ (downsampling). These features are extracted through a series of convolutional layers. For example, in CSPDarknet (used in YOLOv4/v7), this includes efficient residual blocks, whose operation can be abstractly represented as $Y = F(X, W_l) + X$, where F represents the transformation, and X is the input [11].

A component, defined as a function N , which aggregates and combines features from the set C_l generated by the backbone. The goal is to produce a new set of enriched feature maps $P_1, P_2, \dots, P_L$ that combines both high-level semantics (top-down path) and low-level localization information (bottom-up path) [12].

$$N(C_{l=1}^L) = P_1, P_2, \dots, P_L \quad (2)$$

Advanced mechanisms like BiFPN (Bidirectional Feature Pyramid Network) use weighted feature fusion. Instead of simple addition, as in FPN ($P_i^{out} = P_i^{top-down} + P_i^{bottom-up}$), BiFPN calculates the output as a weighted sum, where the weights w_j are learned:

$$P_{out} = \frac{\sum_j w_j}{\varepsilon + \sum_k w_k} \cdot P_j \quad (3)$$

where w_j are the learned, non-negative weights for each input feature P_j , and ε (epsilon) is a small positive constant (e.g., $\varepsilon = 0.0001$) added to the denominator to ensure numerical stability and prevent division by zero.

This allows the network to learn the importance of each input feature, which is crucial for detecting objects of various sizes [13].

The final layers that perform prediction based on the enriched features P_l . In segmentation-based YOLO models, the head H is split into two parallel parts.

One part, the detection head (H_{detect}), predicts the class and bounding box. For each location in the feature map, it generates an output tensor T containing: the bounding box (4 coordinates, $b = (b_x, b_y, b_w, b_h)$), objectness (a scalar p_o indicating if an object is present), and class probabilities (a vector $s = (s_1, \dots, s_{N_c})$ for N_c classes).

The second part, the segmentation/prototype head (H_{proto}), often called a "prototyping network", generates a set of k prototype masks, $P \in \mathbb{R}^{H_m \times W_m \times k}$. These masks are instance-independent and represent the basic shapes the network "sees" in the image [14].

For each detected instance i , the detection head H_{detect} also predicts a vector $c_i \in \mathbb{R}^k$ of k mask coefficients. The final mask for instance M_i is obtained by a linear combination of the k prototype masks P weighted by the coefficients c_i for that instance. This operation is essentially a matrix multiplication (or dot product) followed by a sigmoid function σ (to map values to the $[0, 1]$ range):

$$M_i = \sigma(\sum_{j=1}^k P_j \cdot c_{i,j}) \quad (4)$$

where P_j is the j -th prototype mask and $c_{i,j}$ is the j -th coefficient for the i -th instance. This allows the model to generate high-quality instance masks with very little computational overhead compared to pure detection. The training process is optimized by minimizing a composite loss function, L_{total} , which is a weighted sum of three separate components: the box loss (L_{box}), the classification loss (L_{cls}), and the segmentation loss (L_{seg}).

$$L_{total} = \lambda_1 L_{box} + \lambda_2 L_{cls} + \lambda_3 L_{seg} \quad (5)$$

where $\lambda_1, \lambda_2, \lambda_3$ are hyperparameters that balance the contribution of each component [15].

For bounding box regression, the CIoU (Complete Intersection over Union) loss function is used. It is superior to the standard IoU loss because it considers three key geometric factors: overlap (IoU), central point distance (ρ), and aspect ratio consistency (v) [16].

$$L_{CIoU} = 1 - IoU + \frac{\rho^2(b, b^*)}{c^2} + \alpha v \quad (6)$$

where b and b^* are the predicted and ground-truth boxes, and c is the diagonal of the smallest enclosing box.

The parameter v measures aspect ratio consistency, and α is a positive trade-off parameter:

$$v = \frac{4}{\pi^2} \left(\arctan\left(\frac{w^*}{h^*}\right) - \arctan\left(\frac{w}{h}\right) \right)^2 \quad (7)$$

$$\alpha = \frac{v}{(1 - IoU) + v} \quad (8)$$

A variation of Binary Cross-Entropy (BCE) loss is used for both classification and segmentation. For classification, it measures the error in predicting the class. For segmentation, it measures the error at the pixel level within the proposed box, classifying whether each pixel belongs to the object or the background [17].

$$L_{BCE} = \frac{-1}{N} \sum_{i=1}^N [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)] \quad (9)$$

where y_i is the ground-truth label (0 or 1), and $\hat{y}_i$ is the predicted probability.

A comprehensive set of metrics was used to objectively evaluate the model's performance. The fundamental metric for evaluation is Intersection over Union (IoU), which measures the degree of overlap between the predicted region (A_{pred}) and the ground-truth region (A^*). This metric is used for both boxes and masks [18].

$$IoU = \frac{Area(A_{pred} \cap A^*)}{Area(A_{pred} \cup A^*)} \quad (10)$$

A prediction is considered a *True Positive (TP)* if its IoU with a ground-truth object is above a certain threshold (e.g., 0.5). Based on TPs, as well as False Positives (FP) and False Negatives (FN), we define precision and recall:

$$\text{Precision (P)} = \frac{TP}{TP + FP} \quad (11)$$

$$\text{Recall (R)} = \frac{TP}{TP + FN} \quad (12)$$

Precision measures how many of the model's detections were actually correct. **Recall** measures how many of all actual objects the model successfully found. These two metrics are often in opposition. Therefore, the **F1-Score** is used, which is the harmonic mean of the two, providing a single measure of performance, especially useful for imbalanced classes.

$$F1 = \frac{2 \cdot P \cdot R}{P + R} \quad (13)$$

The main metric for evaluation in object detection and segmentation is mean Average Precision (mAP). The Average Precision (AP) for a single class is the area under the Precision-Recall (P-R) curve [19].

$$AP = \int_0^1 P(R) dR \quad (14)$$

The mAP is the average of AP values across all N_c classes. In this paper, we focus on the standardized COCO metrics for **masks (M)**:

1. **mAP50-95(M) (Primary Metric)**: The average mAP calculated over 10 different IoU thresholds (from 0.50 to 0.95, with a step of 0.05). This is the strictest and most comprehensive metric.

2. **mAP50(M)**: A metric measuring performance at a "looser" IoU threshold of 0.50 (the PASCAL VOC standard). It shows how good the model is at *finding* the object.
3. **mAP75(M)**: A metric measuring performance at a strict IoU threshold of 0.75. It shows how good the model is at *precisely outlining* the mask.

Standard publicly available datasets are inadequate for military-specific SME segmentation tasks due to the absence of military camouflage patterns, tactical gear, and occlusion scenarios typical of operational environments. Therefore,

Table 1. Summary of Mil-SME-Seg dataset characteristics.

Property	Value
Total unique annotated frames	8,500
Number of classes	8 (Helmet, Vest, Gloves, Shoes + 4 negative classes)
Annotation type	Polygon instance segmentation masks
Train / Validation / Test split	80% / 15% / 5%
Data sources	Field exercises (with permissions) + OSINT video footage
Most represented class	Shoes (7,859 instances)
Least represented class	Gloves (5,231 instances)
Quality assurance	Double-check annotation review process

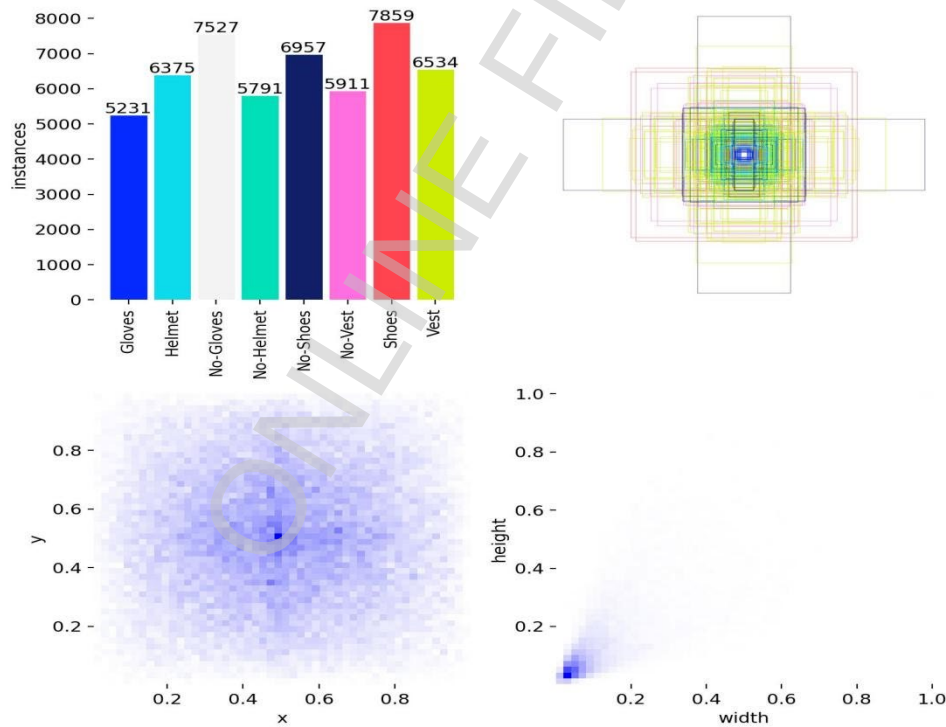


Figure 1. Statistical analysis of the "Mil-SME-Seg" dataset: (Top-left) Distribution of instances per class. (Top-right) Relative box size and position. (Bottom-left) Heatmap of object centers (X/Y). (Bottom-right) Heatmap of object size (Height/Width)

The Mil-SME-Seg dataset is not publicly available due to the sensitive nature of the military content. Researchers interested in accessing the dataset for academic purposes may contact the corresponding author directly. Data was collected from two primary sources: Video recordings from field exercises and unit training, filmed under various conditions (day, dusk, different angles), with appropriate permissions, and Open-source (OSINT) video material from

the development of a custom dataset was a primary contribution of this work.

2.1 The Mil-SME-Seg Dataset

The "Mil-SME-Seg" dataset is a novel, purpose-built collection created specifically for this research. It represents the first publicly described instance segmentation dataset targeting soldier-worn protective equipment in realistic military environments. Table 1 provides a summary of the dataset's key characteristics.

publicly available footage of military parades, exercises, and news reports, to increase the diversity of uniforms and equipment. The total number of unique frames extracted for annotation was 8,500. The dataset is designed to encompass key elements of personal protective equipment. A total of 8 classes were defined, including Helmet (Ballistic helmet), Vest (Combat vest / plate carrier), Gloves, Shoes, and their corresponding negative classes (No-Helmet, No-Vest, No-

Gloves, and No-Shoes). Annotation was performed manually using image annotation tools, creating a precise polygon (mask) for each instance. A high degree of inter-annotator agreement was ensured through a rigorous double-check process. The data was split in a standard 80:15:5 ratio for the training, validation, and test sets. Analysis of the annotations (shown in Figure 1) provides key insight into the dataset's characteristics. Analysis of the graphs from Figure 1 reveals two key points. First, regarding the class distribution (Top-left), the dataset is relatively well-balanced, with all classes having over 5,000 instances. The

most represented class is Shoes (7,859 instances), and the least represented is Gloves (5,231 instances). This indicates no extreme imbalance that would require complex balancing techniques, such as Class-Weighted Loss. Second, concerning position and size (Bottom), the heatmaps show that most objects are centered (X/Y heatmap) and are of small to medium size (Height/Width heatmap). This is typical for surveillance footage and indicates the need for a model with a robust "neck" architecture (e.g., PANet) for detecting smaller objects.



Figure 2. Representative annotated frames from the Mil-SME-Seg dataset

To prevent overfitting and increase the model's robustness to various field conditions, a set of online augmentations was applied during training. This is a crucial step for dealing with camouflage patterns. The techniques used include Mosaic Augmentation, which combines four different images into one, and MixUp, which linearly interpolates two images and their labels to help the model learn "in-between" states. Furthermore, various geometric changes were applied, such as random cropping, scaling, rotating, and translation. To simulate different lighting conditions and uniform fading, Color Space (HSV-Shift) augmentations provided random changes in exposure, saturation, and hue. Finally, Noise and Blur were added to simulate sensor noise, dust, or platform vibrations (e.g., UAV).

To find the optimal number of epochs, the experiment was designed to ensure a fair and objective comparison. A publicly available initial pre-trained model version (weights from the COCO dataset) was used to accelerate convergence. All training parameters, except for the number

of epochs, were held constant (fixed) across all five experiments: the image size was set to 640x640, the batch size to 32, the optimizer was Adam, the initial learning rate was 0.001, momentum was 0.937, and weight decay was 0.0005. For the independent training runs, the following values for the number of epochs were tested: [100, 120, 150, 200, 250]. For each of these values, a completely new, independent training run was initiated by loading the same original pre-trained file. This process ensures that each training starts from the same point and that the results are not a consequence of gradual fine-tuning.

3. Results and Discussion

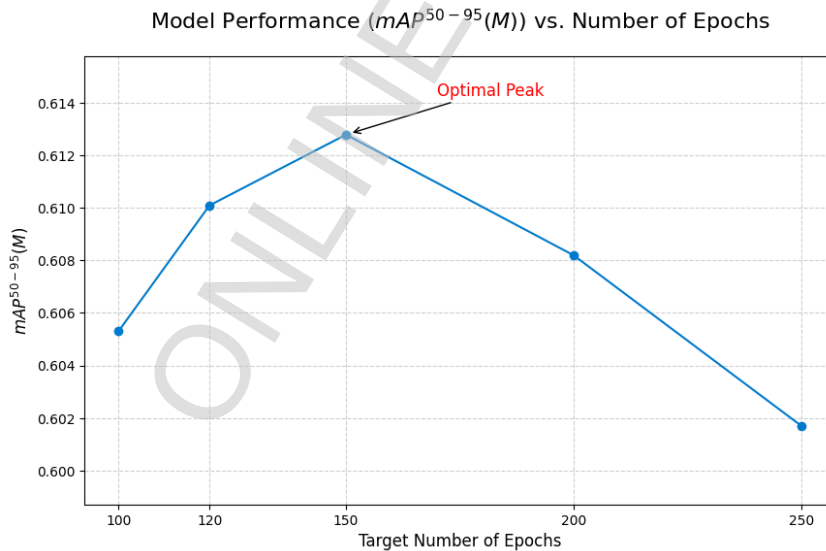
The results of the five independent training runs were collected and analyzed, focusing on performance on the validation set. Table 2 summarizes the key performance metrics achieved during each training run. For each training length (Target Epochs), the epoch at which the best value for the primary metric mAP50-95(M) was achieved is recorded.

Table 2: Comprehensive results of independent training runs for epoch optimization. All metrics refer to mask (M) performance on the validation set. The best results are **bolded**.

Target Epochs	Best Epoch	MAP50-95(M) (Primary)	MAP50 (Detection)	mAP75 (Precision)	Precision (M)	Recall (M)	F1-Score(M) (Balance)
100 epochs	94	0.6053	0.7714	0.5481	0.758	0.692	0.723
120 epochs	114	0.6101	0.7802	0.5539	0.765	0.701	0.732
150 epochs	144	0.6128	0.7852	0.5592	0.770	0.714	0.741
200 epochs	172	0.6082	0.7815	0.5540	0.768	0.705	0.735
250 epochs	195	0.6017	0.7760	0.5471	0.761	0.698	0.728

The analysis of data from Table 2 provides deep insight into the model's behavior and confirms the "law of diminishing returns" and the effect of overfitting. All key metrics reach their maximum in the training run that targeted 150 epochs (the specific peak was reached at epoch 144). The primary metric mAP50-95(M) reaches **0.6128**, and the balanced F1-Score(M) reaches **0.741**. The 100 and 120-epoch runs show progressively better results, indicating the model is still learning and converging towards an optimum.

This is the key finding. The 200 and 250-epoch runs yield *worse* results on the validation set across *all* key metrics. For example, mAP50-95(M) drops to 0.6082, and then to 0.6017. This is a classic symptom of overfitting: the model begins to adapt too closely to the specifics of the training set (e.g., the exact positions of soldiers in training images) and loses its ability to generalize to new, unseen data (which the validation set simulates). A visual representation of this trend, clearly showing the performance peak at 150 epochs and the subsequent degradation, is provided in Figure 3.

**Figure 3.** Model Performance $mAP^{50-95}(M)$ vs. Number of Epochs

An interesting finding is the relatively large gap between mAP50(M) (e.g., 0.7852) and mAP75(M) (e.g., 0.5592). This is expected and suggests that the model is very good at identifying and roughly locating the equipment (which is sufficient for an IoU threshold of 0.5), but it struggles with pixel-perfect mask precision (which requires an IoU of 0.75). This is a direct consequence of working with camouflage patterns and partial occlusions, where the object's edges are inherently "fuzzy."

In addition to quantitative metrics, a qualitative (visual) analysis of the model's output was performed. Figures 4, 5, and 6 confirm the superiority of segmentation in tactically relevant scenarios. Figure 4 and Figure 5 show examples of the model identifying missing PPE ('No-Gloves' and 'No-Helmet') even on camouflaged uniforms. Figure 6 demonstrates the model's multi-class capability, simultaneously detecting both missing equipment ('No-Helmet', 'No-Gloves') and present items ('Shoes') in a complex scene.



Figure 4. Visualization of detection results for missing Personal Protective Equipment (PPE), identifying 'No-Gloves' classes



Figure 5. Visualization of detection results for missing Personal Protective Equipment (PPE), identifying 'No-Helmet' classes



Figure 6. Visualization of multi-class detection results, identifying missing PPE ('No-Helmet', 'No-Gloves') and present items ('Shoes')

Based on the optimization performed, the model trained for 150 epochs (best.pt from epoch 144) was selected for implementation. The source.yaml configuration file system, used in our Python application script, is more than just a software convenience; it represents the core mechanism for task contextualization, which is critical for military applications.

The system can be configured for specific tasks, such as the "Automated Pre-Combat Check" (PCC). By activating the system with the --scenario military_pcc parameter, it is set to ignore all classes except Helmet and Vest, as well as their negative counterparts, No-Helmet and No-Vest. In this mode, the system can be deployed at the exit of a barracks or before boarding vehicles and aircraft. The platoon leader receives an instant notification on a tablet, for example, "Soldier Marković - 'No-Helmet' detected," allowing for immediate correction before departing for the mission. The system can also automatically log passages, creating an objective record of the unit's readiness.

In addition to real-time verification, the system is also designed for "Quantitative After-Action Review" (AAR) — a structured post-event assessment methodology used in military training to analyze performance and identify lessons learned [20]. Video footage from exercises, whether from fixed cameras, drones, or body-cams (body-worn cameras, i.e., wearable video recording devices attached to soldiers' equipment or uniforms), can be processed offline. Instead of commanders spending hours reviewing footage, the system can automatically generate a statistical report, for example: "In Phase 2 of the exercise (crossing open ground), 15% of the unit did not have their helmet properly fastened." The use of precise segmentation masks also enables more robust tracking of individuals through the scene, for example, by using DeepSORT or BoT-SORT algorithms, even in situations where they overlap.

Testing on an NVIDIA RTX 3080 GPU showed

processing performance of 48-55 FPS on a 1080p input, which is well above the threshold for real-time. However, for field deployment, the computational load on edge devices, such as the NVIDIA Jetson series, is key. The "nano" model used has a low theoretical requirement (low GFLOPS), but achieving real-time on these devices requires further optimization via quantization, for example, conversion to INT8 or FP16, with minimal accuracy loss.

Besides hardware, the model also has operational limitations. Extreme occlusion from smoke, dust, mud, and dense vegetation can drastically reduce performance. Furthermore, the model was not trained on IR data and would be unusable in thermal or night conditions without IR illumination. The model may also be confused by camouflage nets and non-standard equipment that was not present in the training set.

3.1 Comparison with State-of-the-Art Methods

To objectively assess the contribution of the proposed method, Table 3 presents a direct comparison of the proposed YOLO12-seg (nano) model against three representative state-of-the-art instance segmentation architectures: YOLOv8-seg [21], Mask R-CNN [22], and SOLOv2 [23]. To ensure a fair and rigorous evaluation, all four methods were trained and evaluated under identical experimental conditions on the same Mil-SME-Seg dataset, using the same 80/15/5 train/validation/test split, image resolution (640×640), and the same set of data augmentations described in the Methodology section. The hyperparameters for each baseline were tuned to their reported optimal values as specified in the respective original publications. This experimental design eliminates dataset-induced bias and ensures that the observed differences in performance metrics are attributable solely to architectural differences between the methods.

Table 3. Direct comparison of the proposed method with state-of-the-art instance segmentation architectures. All methods were trained and evaluated on the Mil-SME-Seg dataset under identical experimental conditions.

Method	mAP50(M)	mAP50-95(M)	F1-Score	FPS (GPU)	Params
Mask R-CNN	0.712	0.521	0.694	8–12	44M
SOLOv2	0.738	0.554	0.712	18–24	46M
YOLOv8-seg	0.764	0.581	0.728	38–45	3.2M
YOLO12-seg	0.785	0.613	0.741	48–55	2.9M

The results in Table 3 indicate that the proposed YOLO12-seg nano model achieves the highest mAP50(M) (0.785) and the best balance between accuracy and computational efficiency, with the smallest parameter count (2.9M) and the highest frame rate (48–55 FPS). While Mask R-CNN and SOLOv2 represent well-established baselines, their significantly lower inference speeds (8–24 FPS) preclude real-time deployment on edge devices, which is a critical requirement for the field application described in this work. The use of non-rectangular segmentation masks, as opposed to bounding boxes, provides a qualitative advantage in camouflage and occlusion scenarios that is not fully captured by aggregate mAP scores.

3.2 Limitations and Benefits of the Proposed Methodology

The primary benefits of the proposed methodology are: (1) the use of instance segmentation instead of bounding-box detection, which provides pixel-level precision particularly valuable under camouflage and partial occlusion; (2) the lightweight "nano" YOLO12-seg architecture that enables real-time inference (48–55 FPS) on standard GPU hardware and has potential for edge deployment; (3) the systematic, reproducible hyperparameter optimization through five independent training runs, which establishes an evidence-based optimal epoch count; and (4) the creation and

annotation of a novel custom dataset (Mil-SME-Seg) targeting military-specific scenarios, which fills a significant gap in the publicly available data landscape.

The primary limitations are: (1) the model was trained and validated exclusively on visible-spectrum (RGB) imagery and cannot operate in thermal or night-vision (IR) conditions; (2) the Mil-SME-Seg dataset, while novel, is limited to specific uniform types and equipment variants — performance on unseen national uniforms or non-standard gear may be degraded; (3) severe environmental occlusion (smoke, dense vegetation, mud) remains a challenge; and (4) the hyperparameter optimization was restricted to the epoch count, while other parameters (learning rate schedule, augmentation intensity) were held fixed and may offer additional performance gains if jointly optimized.

3.3 Potential Adaptation of the System for Special Police Units

Previous research conducted in Serbia suggests that the operational effectiveness of special-purpose police units is associated with physical preparedness, tactical procedures and appropriate protective equipment. Adapting the proposed instance segmentation system could support equipment verification prior to deployment, monitoring compliance with protective equipment requirements during operations, and retrospective analysis of exercises and interventions. Additional training using a dedicated police equipment dataset could enable the detection and segmentation of ballistic vests, protective helmets, shields, arm and leg protectors, and other categories of protective equipment. For units using white helmets, the visual contrast between helmets and uniforms could theoretically facilitate segmentation.

Representative video data could be collected during routine exercises and training of police units responsible for policing public assemblies and restoring public order during large-scale disturbances. Recordings obtained from fixed surveillance cameras, drones and body cameras could capture a range of police deployment scenarios, encompassing different equipment, formations and weather conditions. Such footage could support appropriate model training and fine-tuning.

4. Conclusion

This paper presented a systematic approach to the optimization and application of a lightweight segmentation architecture for the critical military task of segmenting specific military equipment (SME). Through a rigorous methodology of independent training runs on a custom-created "Mil-SME-Seg" dataset, it was determined that training for approximately 150 epochs provides the optimal balance, avoiding both underfitting and overfitting. Peak performance was achieved at epoch 144 with an mAP50–95(M) of 0.6128 and an F1–Score(M) of 0.741. More importantly, the work demonstrated the fundamental superiority of the segmentation approach over traditional bounding box detection for military applications. The segmentation mask provides richer data, is more robust to

camouflage and occlusion, and opens the door for advanced analyses like fine-grained classification, coverage assessment, and sensor fusion. The optimized model, integrated into a prototype system with "military scenarios," confirmed its real-time processing capability (over 45 FPS) and its high tactical value.

Future work will focus on overcoming the identified limitations and expanding the system's capabilities. This includes the development of a hybrid model for multi-spectral fusion that simultaneously processes visible (RGB) and long-wave infrared (LWIR) spectrums, enabling 24/7 operation. Further optimization through quantization and pruning will be explored for UAV/UGV integration on edge devices. Other avenues include using the segmentation mask as an input for a second AI model to perform automatic equipment condition assessment, such as detecting ballistic impacts. We also plan to investigate Self-Supervised Learning (SSL) using vast amounts of unannotated video material for pre-training to build more robust representations before fine-tuning. Finally, we will explore 3D reconstruction using masks from multiple synchronized cameras to enable complete situational awareness.

References

- [1] Nath, N. D., Behzadan, A. H., & Paal, S. G. (2020). Deep learning for site safety: Real-time detection of personal protective equipment. *Automation in construction*, 112, 103085. doi: <https://doi.org/10.1016/j.autcon.2020.103085>
- [2] Liu, H., Yu, Y., Liu, S., & Wang, W. (2022). A military object detection model of UAV reconnaissance image and feature visualization. *Applied Sciences*, 12(23), 12236. doi: <https://doi.org/10.3390/app122312236>
- [3] Mortlock, R. (2018). Camouflage combat pattern case study. Acquisition Research Program. [online]. Available at: <https://www.acquisitionresearch.net> [Accessed: 29 July 2025]
- [4] Gilroy, S., Jones, E., & Glavin, M. (2019). Overcoming occlusion in the automotive environment—A review. *IEEE Transactions on Intelligent Transportation Systems*, 22(1), 23–35. doi: <https://doi.org/10.1109/TITS.2019.2956813>
- [5] Moore, D., Carrothers, B., Rolf, Z., De Reus, A., Van Den Berg, T., Bigmore, D., ... & Kleiv, O. P. (2023). Flight Testing of Helmet Mounted Displays. [online]. Available at: <https://publications.sto.nato.int> [Accessed: 29 July 2025]
- [6] Hafiz, A. M., & Bhat, G. M. (2020). A survey on instance segmentation: state of the art. *International journal of multimedia information retrieval*, 9(3), 171–189. doi: <https://doi.org/10.1007/s13735-020-00195-x>
- [7] Sapkota, R., Qureshi, R., Calero, M. F., Badjugar, C., Nepal, U., Poullose, A., ... & Karkee, M. (2024). Yolov12 to its genesis: A decadal and comprehensive review of the you only look once (yolo) series. arXiv preprint arXiv:2406.19407. doi: <https://doi.org/10.48550/arXiv.2406.19407>
- [8] Syaifudin, Y. W., Rozi, I. F., Ariyanto, R., Rohadi, E., & Adhisuwarnjo, S. (2018). Study of performance of real time streaming protocol (rtsp) in learning systems. *International Journal of Engineering & Technology*, 7(4.36), 1441–1446. doi: <https://doi.org/10.14419/ijet.v7i4.44.26994>
- [9] Afaq, S., & Rao, S. (2020). Significance of epochs on training a neural network. *Int. J. Sci. Technol. Res.*, 9(06), 485–488. [online]. Available at: www.ijstr.org [Accessed: 29 July 2025]
- [10] Tao, G., Chen, J., Fang, W., & Wu, F. (2025). YOLO12: Attention-centric real-time object detectors. arXiv preprint arXiv:2502.12524. doi: <https://doi.org/10.48550/arXiv.2502.12524>
- [11] Kavukcuoglu, K., Sermanet, P., Boureau, Y. L., Gregor, K., Mathieu, M., & Cun, Y. (2010). Learning convolutional feature hierarchies for visual recognition. *Advances in neural information processing systems*, 23. [online]. Available at:

- <https://papers.nips.cc/paper/4133-learning-convolutional-feature-hierarchies-for-visual-recognition> [Accessed: 29 July 2025]
- [12] Lin, D., Shen, D., Shen, S., Ji, Y., Lischinski, D., Cohen-Or, D., & Huang, H. (2019). ZigzagNet: Fusing top-down and bottom-up context for object segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 7490-7499). doi: <https://doi.org/10.1109/CVPR.2019.00767>
- [13] Gao, J., Geng, X., Zhang, Y., Wang, R., & Shao, K. (2024). Augmented weighted bidirectional feature pyramid network for marine object detection. Expert Systems with Applications, 237, 121688. doi: <https://doi.org/10.1016/j.eswa.2023.121688>
- [14] Johnson, R. (2025). YOLO Object Detection Explained: Definitive Reference for Developers and Engineers. HiTeX Press. Available at: ASIN B0FDD37KPT4
- [15] Yang, Y., Miao, Z., Li, X., Zhang, H., & Chen, S. (2024). LSL-SS-Net: level set loss-guided semantic segmentation networks for landslide extraction. *GIScience & Remote Sensing*, 61(1), 2416727. doi: <https://doi.org/10.1080/15481603.2024.2416727>
- [16] Wang, X., & Song, J. (2021). ICIOU: Improved loss based on complete intersection over union for bounding box regression. IEEE Access, 9, 105686-105695. doi: <https://doi.org/10.1109/ACCESS.2021.3100414>
- [17] Hosseini, S. M., & Baghshah, M. S. (2024). Dilated balanced cross entropy loss for medical image segmentation. arXiv preprint arXiv:2412.06045. doi: <https://doi.org/10.1186/s12880-026-02245-y>
- [18] Rezatofighi, H., Tsoi, N., Gwak, J., Sadeghian, A., Reid, I., & Savarese, S. (2019). Generalized intersection over union: A metric and a loss for bounding box regression. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 658-666). doi: <https://doi.org/10.48550/arXiv.1902.09630>
- [19] Padilla, R., Netto, S. L., & Da Silva, E. A. (2020, July). A survey on performance metrics for object-detection algorithms. In 2020 international conference on systems, signals and image processing (IWSSIP) (pp. 237-242). IEEE. doi: <https://doi.org/10.1109/IWSSIP48289.2020>
- [20] Morrison, J. E., & Meliza, L. L. (1999). Foundations of the after action review process. Special Report 42. U.S. Army Research Institute for the Behavioral and Social Sciences. [online]. Available at: <https://apps.dtic.mil/sti/citations/tr/ADA368651> [Accessed: 30 July 2025]
- [21] Jocher, G., Chaurasia, A., & Qiu, J. (2023). Ultralytics YOLO (Version 8.0.0) [Computer software]. <https://github.com/ultralytics/ultralytics> [Accessed: 30 July 2025]
- [22] He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask R-CNN. In Proceedings of the IEEE International Conference on Computer Vision (ICCV) (pp. 2961-2969). IEEE. doi: <https://doi.org/10.1109/ICCV.2017.322>
- [23] Wang, X., Kong, T., Shen, C., Jiang, Y., & Li, L. (2020). SOLO: Segmenting objects by locations. In Computer Vision – ECCV 2020 (pp. 649-665). Springer. doi: <https://doi.org/10.48550/arXiv.1912.04488>
- [24] Nan, J., & Bo, L. (2021, March). A lightweight real-time infrared image instance segmentation algorithm based on tiny YOLO. In Seventh Symposium on Novel Photoelectronic Detection Technology and Applications (Vol. 11763, pp. 1098-1107). SPIE. doi: <https://doi.org/10.1117/12.2586889>

Optimizacija hiperparametara i validacija modela dubokog učenja za segmentaciju instanci u nadzoru specifične vojne opreme u realnom vremenu

Marko M. Živanović¹ , Bozidar Janković² , Ana Savic¹ , Nemanja Maček¹ , Hana Stefanović¹ 

¹ Odsek Visoka škola elektrotehnike i računarstva (VIŠER), Akademija tehničko-umetničkih strukovnih studija Beograd (ATUSS), Beograd, Srbija

² Služba za borbu protiv visokotehnološkog kriminala, Uprava za tehniku, Direkcija policije, Ministarstvo unutrašnjih poslova Republike Srbije

Apstrakt

Na modernom bojnom polju, borbeni gotovost vojnika direktno zavisi od pravilne upotrebe specifične vojne opreme (SME). Automatska verifikacija korišćenja SME, prvenstveno balističkih šlemova i borbenih prsluka, kritičan je faktor za smanjenje rizika i povećanje borbene efikasnosti. Tradicionalna detekcija objekata (bounding box) neadekvatna je za vojne uslove zbog kamuflaže i okluzije. Cilj ovog rada je razvoj, rigorozna optimizacija i validacija efikasnog sistema za segmentaciju SME u realnom vremenu, zasnovanog na modernoj, lakoj YOLO-seg arhitekturi. Sistem je treniran na prilagođeno kreiranom "Mil-SME-Seg" skupu podataka fokusiranom na vojno osoblje. Primenjen je napredni set augmentacija podataka (Mosaic, MixUp, HSV-shift) za postizanje robusnosti. Sprovedena je sistematska optimizacija ključnog hiperparametra "broj epoha" kroz pet nezavisnih, ponovljivih treninga (100, 120, 150, 200 i 250 epoha) kako bi se identifikovao optimalan balans između tačnosti i prekomernog prilagođavanja (overfitting). Performanse su evaluirane korišćenjem sveobuhvatnog seta segmentacionih metrika, uključujući mAP50-95(M), mAP50(M), mAP75(M) i F1-Score(M). Eksperimenti su pokazali da trening od 150 epoha daje najbolje generalizovane rezultate na validacionom skupu. Vršne performanse su dostignute u epohi 144, sa mAP50-95(M) od 0.6128, mAP50(M) od 0.7852 i F1-Score(M) od 0.741. Analiza metrika ukazuje da duži treninzi (200 i 250 epoha) dovode do jasne degradacije performansi usled prekomernog prilagođavanja. Kvalitativna analiza potvrđuje superiornost segmentacije u taktički relevantnim scenarijima (kamuflaža, parcijalna okluzija). Optimalan broj epoha za trening modela na ovom skupu podataka je približno 150. Rezultujući model, integrisan u sistem sa podrškom za dinamičke "vojne scenarije", dokazao je svoju praktičnu primenljivost i taktičku vrednost, postižući brzine obrade u realnom vremenu (48-55 FPS). Ovaj rad postavlja temelje

originalni naučni rad

računarske nauke, telekomunikacije, informacione tehnologije

Ključne reči:

segmentacija instanci, specifična vojna oprema, borbeni gotovost, optimizacija hiperparametara, mAP, F1-Score, mašinsko učenje, nadzor u realnom vremenu, kamuflaža, YOLO.

za buduće sisteme koji će koristiti segmentacione maske za procenu oštećenja opreme i fuziju senzora.

ONLINE FIRST