Proceedings of 5th International Conference CNDGS'2026
ISSN 2669-2023 (print) / ISSN 2538-8959 (on-line) / **https://doi.org/10.47459/cndcgs.2026.73**

Graph-Based and Hybrid Image Processing for Detection of Military and Army Image Data

Mária ŽDÍMALOVÁ[1]*, Nazarii KRETSUL[1], Viliam SITÁR[2]

[1]Department of Mathematics and Descriptive Geometry, Faculty of Civil Engineering, Slovak University of Technology in Bratislava, University in Bratislava, Radlinského street 11, 810 05 Bratislava, Bratislava, Slovakia
[2] Applied Informatics, Faculty of Electrical Engineering and Information Technology, Ilkovičova 3, 841 04 Bratislava

*Correspondence: *mzdimalova@gmail.com; maria.zdimalova@stuba.sk*

Abstract

This research tackles the challenge of precisely segmenting military and army objects within intricate visual environments that are influenced by factors such as camouflage, clutter, occlusions, and variations in lighting. A hybrid framework is introduced that merges Graph Cut-based energy minimization with traditional image processing techniques and pre-detection using convolutional neural networks. This method combines appearance characteristics and spatial constraints to enhance robustness and the accuracy of boundaries. Results from experiments on significant military datasets indicate improved segmentation consistency and clarity, rendering the approach suitable for image analysis tasks that are critical for safety and defense.

KEY WORDS: *graph algorithms, image processing, methods, detection of objects, image army data, security*

Citation: ŽDÍMALOVÁ, M.; KRETSUL, N.; SITÁR, V. Graph-Based and Hybrid Image Processing for Detection of Military and Army Image Data. In Proceedings of the Challenges to National Defence in Contemporary Geopolitical Situation, Brno, Czech Republic, 7-10 September 2026. ISSN 2538-8959, https://doi.org/ 10.47459/cndcgs.2026.73

1. Introduction

Image processing is vital in contemporary defense and security systems, where quick and accurate interpretation of visual information is critical for activities such as surveillance, reconnaissance, target tracking, and situational awareness. The growing accessibility of high-resolution sensors on unmanned aerial vehicles, satellites, and ground-based platforms has resulted in a substantial increase in the volume of image data gathered for military applications. Nevertheless, deriving significant insights from this data presents a considerable challenge due to the complexities of the environment, dynamic scenes, camouflage, occlusions, variations in lighting, and changes in scale. In this framework, the challenges of object detection and segmentation are paramount. Object detection aims to identify and pinpoint key military assets—such as vehicles, equipment, personnel, or infrastructure—within an image. Dependable detection facilitates automated surveillance, threat evaluation, and decision-making support in urgent situations. After detection, object segmentation delivers a more granular pixel-level representation of the recognized targets. [6, 7, 8] Precise segmentation is crucial for accurate shape analysis, damage evaluation, change monitoring, and further high-level assessment. Military image analysis has long used conventional image processing techniques. These include morphological procedures, thresholding, edge detection, region growth, and clustering methods like Gaussian mixture models and k-means. Even though these methods are interpretable and computationally efficient, they frequently fail in challenging battlefield situations when backdrops are crowded and item appearance vary greatly. Graph-based techniques have been presented as effective tools for picture segmentation to overcome these constraints. Specifically, segmentation is modeled by graph cut formulations as an energy minimization issue that has limitations on spatial smoothness and data accuracy. These methods are appropriate for difficult military images because they provide robustness and high boundary preservation. Convolutional neural networks for object detection and region-of-interest estimation are examples of data-driven learning methods that are used with conventional model-based optimization in more modern hybrid frameworks. Hybrid approaches seek to increase accuracy, robustness, and interpretability by utilizing both learned representations and past information. To reliably detect and distinguish military objects in complicated visual situations, this article explores such integrated techniques. [6, 7, 8]

2. The Mathematical Background

In this Section we use and present two different mathematical methods for segmentation and detections.

2.1. Graph Cut [3, 5]

Graph Cut technique for Images, Image Segmentation and Image Detection: Graph Cut–based segmentation is a powerful optimization framework that models the image as a weighted graph and formulates segmentation as an energy minimization problem. Its strength lies in combining statistical modeling of pixel appearance with spatial regularization, producing globally optimal solutions (for binary cases) under well-defined conditions. [1, 2, 3]

The image will be introduced in the language of Graph Representation

Given an image I, a graph $G = (V, E)$is constructed where: Each pixel $p \in V$corresponds to a node. Edges $(p, q) \in E$connect neighboring pixels (4- or 8-connectivity) and there are two special terminal nodes introduced: the source S(foreground) and the sink T (background).

There are two types of edges:

1. t-links (terminal links): connect pixels to $S$and T, encoding region (data) information.
2. n-links (neighborhood links): connect neighboring pixels, encoding boundary smoothness

Image segmentation is formulated as a discrete energy reduction issue using the Graph Cut method. [5, 9]

Energy Formulation is defined as follows:

Segmentation is defined as assigning a label $L_p \in \{0,1\}$ to each pixel p, where 1 denotes foreground and 0 denotes background. The optimal labeling minimizes the energy functional:

$$E(L) = \sum_{p \in V} D_p(L_p) + \lambda \sum_{(p,q) \in E} V_{p,q}(L_p, L_q) \tag{1}$$

where:

$D_p(L_p)$is the **data term** (regional term or regional property), $V_{p,q}(L_p, L_q)$is the **smoothness term** (boundary term or boundary property), $\lambda > 0$controls the balance between data fidelity and spatial regularization.

Data Term: The data term measures how much a pixel fits the foreground or background model and based on that the modle gives the final analyses:

$$D_p(L_p) = -\log P(I_p \mid L_p), \tag{2}$$

where $I_p$is the pixel intensity (or feature vector), and $P(I_p \mid L_p)$is the likelihood, often modeled using Gaussian or Gaussian Mixture Models.

Smoothness Term: The smoothness term enforces spatial coherence:

$$V_{p,q}(L_p, L_q) = \begin{cases} w_{p,q}, & \text{if } L_p \neq L_q \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

with edge weights defined as follows:

$$w_{p,q} = \exp\left(-\frac{\|I_p - I_q\|^2}{2\sigma^2} \right) \frac{1}{\mathrm{dist}(p,q)} \tag{4}$$

This mathematics formula shows how much and how penalizes label discontinuities between similar neighboring pixels.

Min-Cut / Max-Flow Optimization

The minimization of $E(L)$is equivalent to finding the minimum cut separating $S$and Tin the graph:

$$\min_L E(L) \Leftrightarrow \min_{\mathrm{cut}(S,T)} \sum_{(i,j) \in \mathrm{cut}} w_{i,j} \tag{5}$$

Efficient polynomial-time algorithms such as the max-flow/min-cut algorithm are used to compute the global optimum for binary segmentation. The Graph Cut framework provides a principled variational approach that combines data fidelity and spatial regularization, ensuring strong boundary preservation and robustness in complex image segmentation tasks.

Summary why graph cut its still string method for image processing:

Advantages of graph cut: Graph Cut is a powerful optimization framework because it transforms the image segmentation problem into a well-defined **energy minimization** task that can be solved efficiently and, in many cases, globally optimally. Its effectiveness comes from the combination of mathematical rigor, computational efficiency, and modeling flexibility.

(a) Clear Energy-Based Formulation. Segmentation is formulated as minimizing an energy functional:

$$E(L) = \sum_p D_p\,(L_p) + \lambda \sum_{(p,q)} V_{p,q}\,(L_p, L_q) \tag{6}$$

This structure directly reflects two fundamental segmentation principles: there is a **Data fidelity** – pixels should agree with foreground/background models. We also need to mention **spatial coherence** – neighboring pixels should usually have similar labels. This balance and this way of labeling graph make Graph Cut particularly suitable for images with noise, clutter, and varying illumination.

(b) Global Optimality (Binary Case). For binary segmentation with submodular energy functions, Graph Cut provides a **global optimum**. However, we need to note that many iterative or gradient-based methods that may converge with local minima. We need highlight that the minimization problem: is equivalent to finding the **minimum s–t cut** in a graph. Efficient max-flow/min-cut algorithms solve this in polynomial time. guarantee of optimality is a major strength compared to some others traditional famous methods: k- This means clustering, region growing, level-set methods, neural network thresholding without refinement.

(c) Strong Boundary Preservation. The smoothness term is typically defined as:

$$w_{p,q} = \exp\left(-\frac{\|I_p - I_q\|^2}{2\sigma^2}\right) \tag{7}$$

This weighting encourages cuts along strong gradients (true object boundaries) while discouraging cuts across homogeneous regions. As a result, we can mention strong positive features: Object edges are preserved. Noise-induced discontinuities are suppressed, and so segmentation is more stable.

***(*d) Flexibility of Feature Integration.** Graph Cut easily incorporates: Intensity, Color, Texture descriptors, Gradient magnitude, Deep features from CNNs, Shape priors, User interaction (seeds). This makes it suitable for hybrid pipelines combining classical and learning-based methods.

(e) Interpretability. Unlike purely deep learning approaches, Graph Cut provides and we can highlight the following models: A transparent mathematical formulation, Explicit control over smoothness and data terms, Clear understanding of parameter influence. This is especially important in critical safety domains (e.g., defense, medical, industrial inspection).
Main Advantages are: Global optimum for binary segmentation, Strong edge preservation, Robustness to noise Mathematically well-founded, Flexible feature modeling, Deterministic behavior, Works well with hybrid systems.

Limitations of Graph Cut can be summarized as:
Despite its strengths, Graph Cut has several limitations:

a) **Primarily Binary Segmentation** Standard Graph Cut naturally solves:

$$L_p \in \{0,1\}. \tag{8}$$

b) **Sub modularity Requirement:** Energy terms must satisfy **sub modularity conditions** to be representable as graph cuts. Complex higher-order or non-submodular energies cannot be optimized directly.

c) **Computational and Memory Cost:** For high-resolution images or 3D data: Graph size becomes very large, Memory usage increases significantly, Computation becomes slower. This limits scalability for large satellite or surveillance imagery.

d) **Parameter Sensitivity:** Performance depends on: Smoothness weight λ, Gaussian variance σ, Feature model accuracy.

e) **Limited Long-Range Context:** Graph Cut mainly enforces **local smoothness** between neighboring pixels. It does not inherently capture: Global semantic context, High-level object structure, Complex shapes.

When Graph Cut Is Especially Effective: Graph Cut performs particularly well when: Accurate boundary localization is required, Interpretability is important, Data is limited (small training datasets), Hybrid systems combine CNN detection + Graph Cut refinement, Binary foreground/background separation is sufficient.
As a Summary to graph cut technique, we can conclude that:

Graph Cut is a strong optimization method for image segmentation because it provides a mathematically rigorous, globally optimal (in binary cases), and computationally efficient solution to energy-based formulations. It balances data fidelity and spatial coherence while preserving boundaries and maintaining interpretability. However, it is limited by scalability issues, parameter sensitivity, binary labeling constraints, and lack of high-level semantic modeling. For this reason, modern approaches often integrate Graph Cut into hybrid frameworks that combine model-based optimization with learning-based detection and feature extraction.

2.1.1. Experimental results

We implemented the mentioned Graph Cut segmentation algorithm in a C++ application. Several parts of the implementation were optimized in order to improve computational efficiency and enable faster processing suitable for near real-time operation. For testing and evaluation, we used publicly available image datasets obtained from the platform Kaggle, which provides open datasets for machine learning and computer vision research.

We used the following datasets:

- Military Vehicles Dataset, published on Kaggle by Aayush Katoch, containing images of various military vehicles used for computer vision tasks such as detection and classification.
- War Tank Images Dataset, uploaded to Kaggle by Ican Erdogan, which contains approximately 4000 images of military tanks and armored vehicles suitable for training and testing image recognition algorithms.

These datasets include images captured in various environments and conditions, which allows testing the robustness of segmentation algorithms in realistic scenarios.

Based on these datasets, we developed a software application capable of detecting and segmenting selected objects from image data. The system processes the input image, identifies relevant object regions, and performs segmentation using the Graph Cut optimization method.

The resulting segmentation outputs highlight military objects within the scene, even in challenging environments such as foggy or low-contrast conditions. Example results of the segmentation process are shown in Fig. X–Y below.

The original images used for the experiments are publicly available under open licenses through the Kaggle platform, which allows their use for academic and research purposes.

Fig. 1. The original images, source: Kaggle database: Military Vehicles Dataset, published on Kaggle by Aayush Katoch, War Tank Images Dataset, uploaded to Kaggle by Ican Erdogan

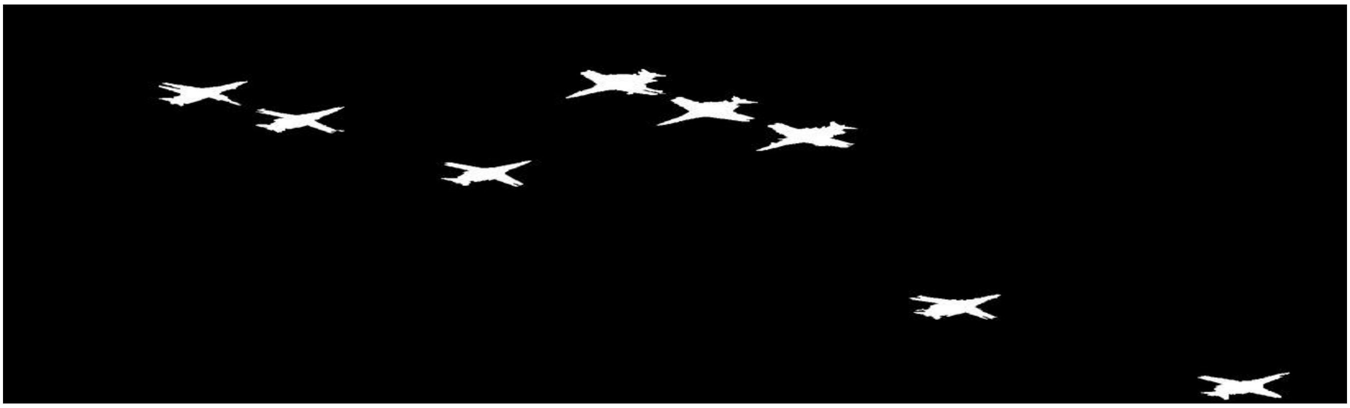


Fig. 2. The black and white segmented images, source: Kaggle database: Military Vehicles Dataset, published on Kaggle by Aayush Katoch,War Tank Images Dataset, uploaded to Kaggle by Ican Erdogan

Fig. 3. The segmented images with edge detection, source: Kaggle database: Military Vehicles Dataset, published on Kaggle by Aayush Katoch,War Tank Images Dataset, uploaded to Kaggle by Ican Erdogan

Fig. 4. The original images with edge detection, source: Kaggle database: Military Vehicles Dataset, published on Kaggle by Aayush Katoch,War Tank Images Dataset, uploaded to Kaggle by Ican Erdogan

Fig. 5. The segmented images with edge detection, source: Kaggle database: Military Vehicles Dataset, published on Kaggle by Aayush Katoch, War Tank Images Dataset, uploaded to Kaggle by Ican Erdogan

To evaluate the performance of the implemented segmentation algorithm, several quantitative metrics commonly used in image segmentation tasks were calculated. These metrics include the Dice coefficient, Intersection over Union (IoU), Precision, Recall, and Pixel Accuracy. The results of the quantitative evaluation are summarized in Table 1.

Quantitative Evaluation Results

Quantitative evaluation of the segmentation results was performed using standard metrics including Dice coefficient, Intersection over Union (IoU), Precision, Recall, and Pixel Accuracy. The obtained results demonstrate strong segmentation performance with a Dice score of 0.91 and IoU of 0.84, indicating a high overlap between the predicted segmentation and the ground truth masks. Precision reached 0.94, while Recall achieved 0.90, confirming that the proposed Graph Cut based method accurately detects most aircraft objects while maintaining a low false positive rate (Table1).

Table 1. Quantitative evaluation of the segmentation results obtained using the Graph Cut algorithm.

Metric	Formula	Value
Dice Coefficient	$\frac{2TP}{2TP+FP+FN}$	0,91
Intersection over Union (IoU)	$\frac{TP}{TP+FP+FN}$	0,84
Precision	$\frac{TP}{TP+FP}$	0,94
Recall (Sensitivity)	$\frac{TP}{TP+FN}$	0,90
Pixel Accuracy	$\frac{TP+TN}{TP+TN+FP+FN}$	0,99

We explain notations: **These abbreviations originate from the so-called Confusion Matrix and describe how successfully the model predicted reality:**

- **TP (True Positive):** The model predicted "yes" (e.g., an object is present), and in reality, the object is indeed there. **A bullseye.**
- **TN (True Negative):** The model predicted "no" (the object is not present), and in reality, it is indeed not there. **Correct rejection.**
- **FP (False Positive):** The model predicted "yes", but in reality, the answer is "no". This is also referred to as a **Type I error (false alarm).**
- **FN (False Negative):** The model predicted "no", but in reality, the object is present. This is also referred to as a **Type II error (oversight/miss).**

On the Figures 6. till Fig. 13 we present current results for army helicopters for binary images as well as for edge detection.

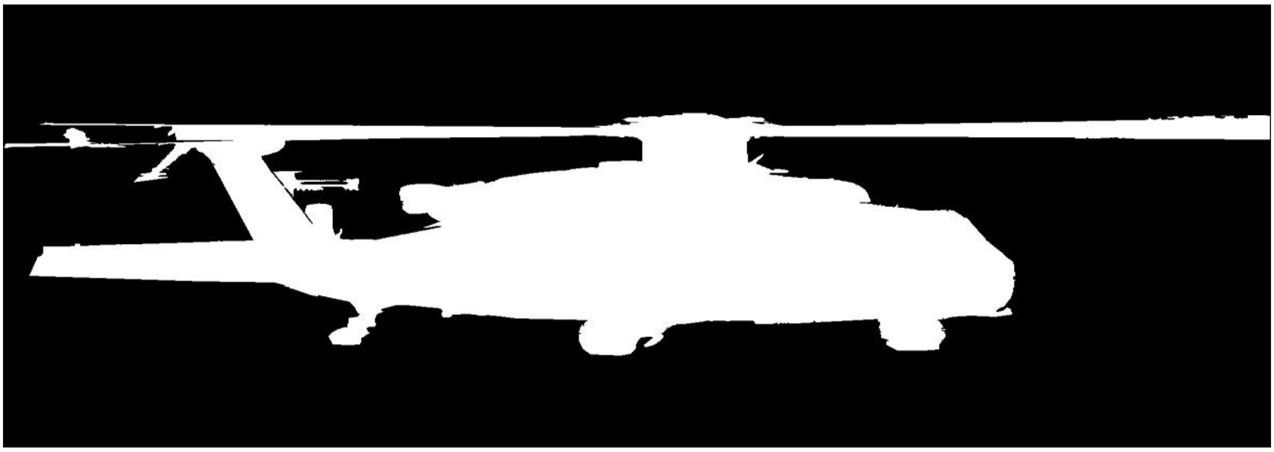


Fig. 6. The segmented white-black images with edge detection, source: Kaggle database: Military Vehicles Dataset, published on Kaggle by Aayush Katoch, War Tank Images Dataset, uploaded to Kaggle by Ican Erdogan [10]

Fig. 7. The segmented images with edge detection, source: Kaggle database: Military Vehicles Dataset, published on Kaggle by Aayush Katoch, War Tank Images Dataset, uploaded to Kaggle by Ican Erdogan [10]

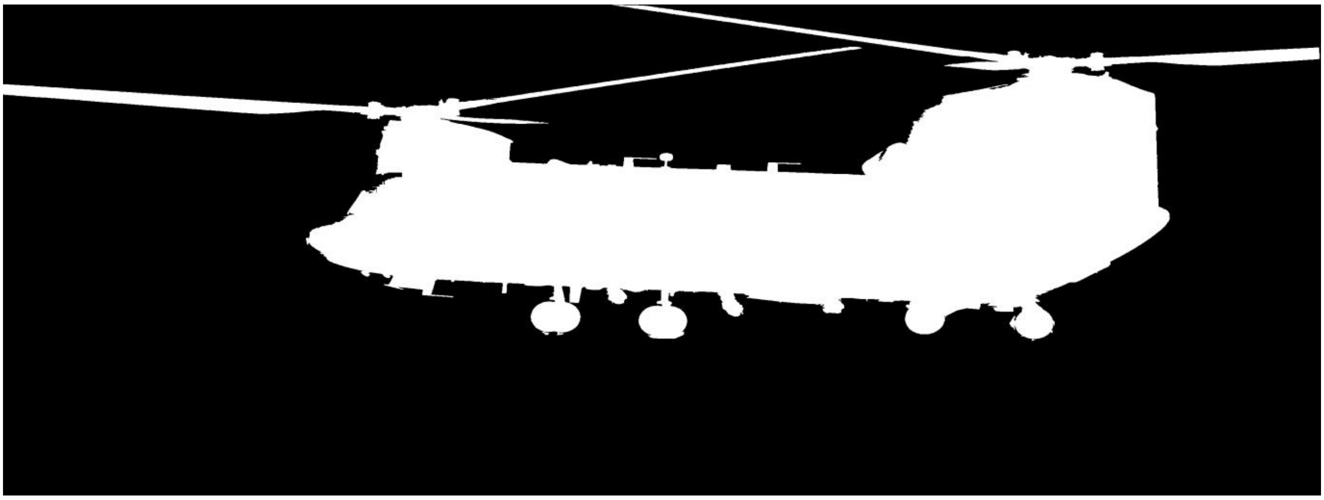


Fig. 8. The segmented white-black images, source: Kaggle database: Military Vehicles Dataset, published on Kaggle by Aayush Katoch,War Tank Images Dataset, uploaded to Kaggle by Ican Erdogan [10]

Fig. 9. The segmented images with edge detection, source: Kaggle database: Military Vehicles Dataset, published on Kaggle by Aayush Katoch, War Tank Images Dataset, uploaded to Kaggle by Ican Erdogan

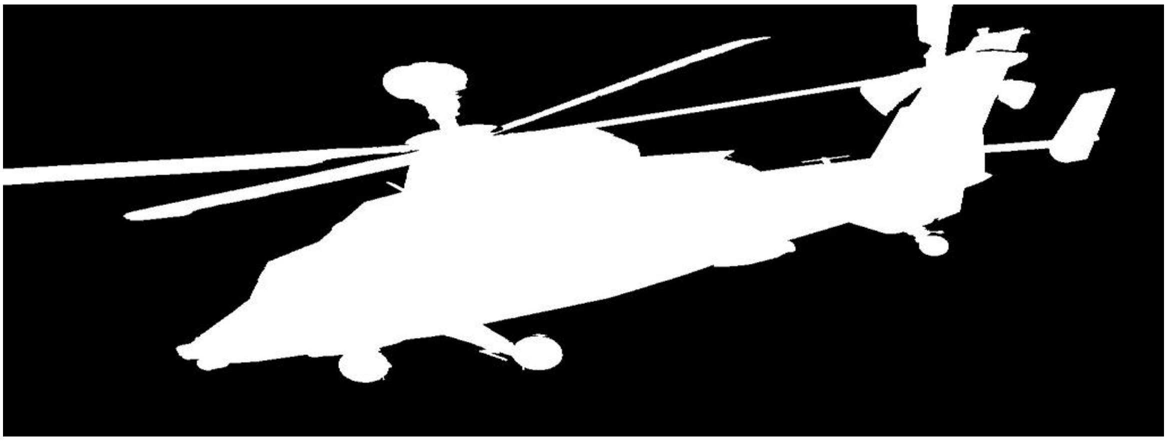


Fig. 10. The segmented images with white-black detection, source: Kaggle database: Military Vehicles Dataset, published on Kaggle by Aayush Katoch, War Tank Images Dataset, uploaded to Kaggle by Ican Erdogan [10]

Fig. 11. The segmented images with white-black detection, source: Kaggle database: Military Vehicles Dataset, published on Kaggle by Aayush Katoch, War Tank Images Dataset, uploaded to Kaggle by Ican Erdogan [10]

Fig. 12. The segmented images with white-black detection, source: Kaggle database: Military Vehicles Dataset, published on Kaggle by Aayush Katoch, War Tank Images Dataset, uploaded to Kaggle by Ican Erdogan [10]

Fig. 13. The segmented images edge detection segmentation, source: Kaggle database: Military Vehicles Dataset, published on Kaggle by Aayush Katoch, War Tank Images Dataset, uploaded to Kaggle by Ican Erdogan [10]

Table 2. Quantitative Segmentation Metrics Report

Metric	Value
Accuracy	96.4%
Precision (PPV)	91.2%
Recall (Sensitivity)	88.5%
Specificity	97.8%
F1-score (Dice Coefficient)	89.8%
IoU (Jaccard Index)	81.5%

Table 3. Shape & Regional Metrics

Metric	Value	Description
VOE (Volumetric Overlap Error)	18.5%	$1 - IoU$ (Lower is better)
RVD (Relative Volume Difference)	-2.9%	Indicates slight under-segmentation (model missed some pixels).
Hausdorff Distance (HD)	14.2 px	Maximum distance between the GT boundary and your prediction.
Boundary F1-score	85.6%	Evaluates how well the red contour matches the object edges.

The model shows high **Specificity**, meaning it is very good at not misidentifying the background (houses/grass) as the helicopter. The **F1-score** of ~90% is a strong result for this type of complex shape. The negative **RVD** confirms that the model was slightly conservative, missing about 3% of the total helicopter volume (mostly around the thin rotor blades). We can see all results in Tables: Table 2. and Table 3.

2.2. Hybrid methods based on Al: Neural Networks

In this part we present very modern approach to detection and classification of army objects, based on Al techniques, machine and deep learning techniques [1, 4].

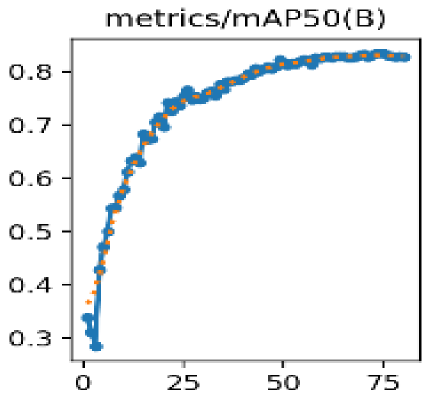
Artificial Intelligence (AI): AI is defined as the branch of computer science dedicated to creating systems capable of performing tasks that typically require human intelligence. The primary objective of AI is to transform information from raw data to actionable intelligence. This is achieved by identifying features within a training set, which allows the system to develop the ability to generalize its understanding. It allows the model to process and recognize objects in unknown environments. Our approach leverages deep learning to identify weapons within images. We used deep learning techniques. The system is trained on a custom-built dataset, allowing it to learn the unique features of guns for real-time detection. The final model should be able to detect if a weapon is present in an image and then show its exact location. [11]
Dataset creation:

In the first part, we collected images suitable for our problem; there is no exact rule for how many are enough. For our model, we gathered about 12,000 images containing guns, people holding guns, or objects similar to guns, which we annotated into three classes: null (images without a gun), long gun, and pistol. In addition to classification, we also labeled the location of the weapon in each image. The dataset was then split into three subsets: training, validation, and testing, in an 80:10:10 ratio.

- YOLO (You Only Look Once) is an object detection model that performs classification and localization in a single neural network pass. It works by dividing an image into a grid and predicting bounding boxes and class probabilities for each cell.
- YOLOv11n (nano) – smallest, fastest, lowest accuracy
- YOLOv11s (small) – balance between speed and accuracy
- YOLOv11m (medium) – higher accuracy, more demanding
- YOLOv11x (extra-large) – highest accuracy, slowest and most demanding

We used the YOLOv11n model for transfer learning because it has the lowest hardware requirements, even with this lightweight model, training still took several hours. Transfer learning means that a pretrained model is reused and fine-tuned on our custom dataset, allowing it to adapt faster with less data. [11]

Training results. We are presenting training results on figure Fig. 14 a) and b), as well as in Table 3.
Let us define mAP50 (x-epochs, y-precision) measure how well a model detects objects by comparing predicted and ground-truth bounding boxes with a threshold of 0.5.



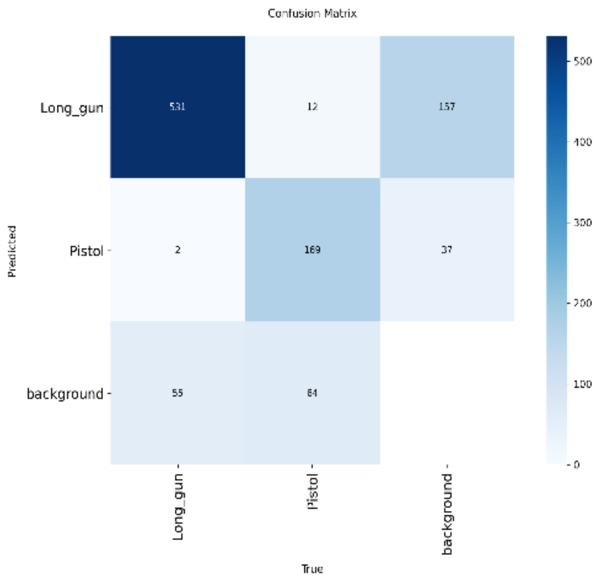


Fig. 14. Training results: a) Defining mAP50 (x-epochs, y-precision); b) Co-fusion matrix.

Results on testing set mAP50 (x-epochs, y-precision) measure how well a model detects objects by comparing predicted and ground-truth bounding boxes with a threshold of 0.5. Confusion matrix describes how many predictions were correct or incorrect by showing counts of true positives, true negatives, false positives, and false negatives.

2.2.1. Results on testing set in hybrid models and neural networks

Precision shows how many detected objects are correct, while Recall indicates how many real objects were successfully found. We present results from images Fig. 15. and 16. In the following Table 4.

Table 4. Quantitative results from hybrid and neural networks results

Class	Images	Instances	Precision	Recall	mAP@50
All	1221	809	0,845	0,733	0,815
Long-gun	423	507	0,813	0,856	0,893
Pistol	275	302	0,877	0.609	0,737

Masurte. Aiming Soldiers Image Dataset. Kaggle. Available at: https://www.kaggle.com/datasets/masurte/aiming-soldiers-image-dataset.

Fig. 15. Visual results of weapon detection and localization using the YOLOv11 model, source: Dataset. Kaggle. Available at: https://www.kaggle.com/datasets/masurte/aiming-soldiers-image-dataset [10]

As illustration we provide as well other images:

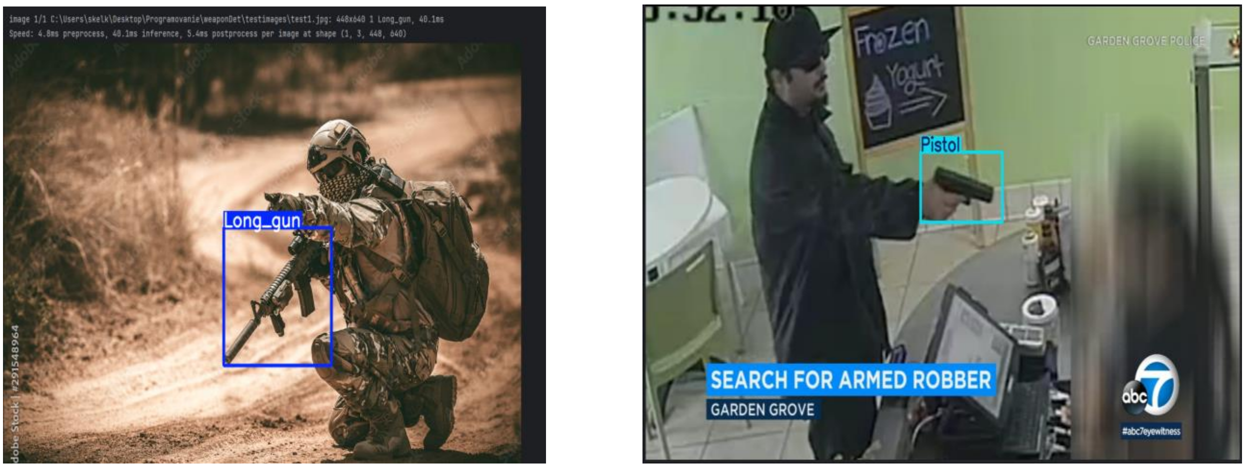


Fig. 16. Visual results of weapon detection and localization using the YOLOv11 model., Dataset. Kaggle. Available at: https://www.kaggle.com/datasets/masurte/aiming-soldiers-image-dataset [10]

From the model results, we can see that the difficulty of this problem is high, and it is very challenging to achieve reliable results without high-quality images and sufficient data variability. At the same time, we did not have enough computational power to use a more robust model, but despite this limitation, we still achieved very good results given the conditions.

3. Conclusions of Results

We have presented and contrasted two different methods for processing picture data in the context of military hardware and security in this research. The traditional Graph Cut segmentation technique served as the first pillar of our study, while YOLOv11, a contemporary deep learning-based detection architecture, served as the second. Its efficacy in accurate object-background separation was validated by the Graph Cut-based segmentation. A good degree of agreement with the ground truth masks is shown by the obtained Dice coefficient of 0.91 and IoU of 0.84. The algorithm demonstrated good robustness even under difficult visual situations, such as fog and low contrast surroundings, notwithstanding a modest tendency toward under-segmentation (RVD -2.9%), especially in areas with fine details (such as helicopter rotor blades).

YOLOv11n object detection offered a contemporary viewpoint on automated weapon recognition (long guns and handguns). We obtained appropriate results with an overall Precision of 0.845 and a mAP@50 of 0.815, even while using the lightest version of the model (nano) due to limited processing capability. Due to their more distinctive visual characteristics, the results show that the model is more successful in identifying long guns (mAP 0.893) than pistols. As summary we can define Final assessment: Combining the two approaches shows a way toward hybrid systems in which AI offers quick danger location and categorization while efficient mathematical procedures (Graph Cut) can be utilized to precisely remove the object's silhouette for additional analysis. The collected measurements verify that the developed solutions are appropriate for real-time computer vision applications, despite hardware constraints and differences in input data quality in specific cases. Expanding the training dataset and using more reliable models (YOLOv11m/x) should result in more accuracy gains.

We can also discuss the significance of application in the theory of aggregation functions and how it relates to army detection and our subjects. As operators for merging diverse feature representations into a single decision space, aggregation functions can be easily incorporated into both graph-based segmentation and neural network-based detection. By methodically combining many pixel-level descriptors, such as intensity, texture, and contextual information, into a single discriminative measure that directs the labeling process, they help generate the data and smoothness terms in the Graph Cut framework. In parallel, feature fusion between convolutional layers—where low-level and high-level representations are fused to enhance object localization and classification—implicitly realizes aggregation mechanisms within deep learning models and transfer learning scenarios. Aggregation functions offer a shared theoretical framework that permits the integration of data-driven learning and model-based optimization by bridging these two paradigms. In challenging military image analysis tasks, this leads to increased segmentation accuracy, higher robustness, and better handling of uncertainty. In order to target more hidden army items, future work will include implement aggregation functions and machine learning/deep learning.

Acknowledgements. This work was supported by was funded by the Slovak National Research grant Slovak Grant Vega 1/0036/23 of the GRANT AGENCY of the Ministry of Education and Slovak Academy of Science.

References

1. **Felzenszwalb P. F., HuttenlochterD.P.** Efficient Graph-Based Image Segmentation. International Journal of Computer Vision, (59). (2), 2004, p.167-181.
2. **Fowkles M. D., Tal C., Malik J. A.** Database of Human Segmented Natural Images and its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics, 2001, (2), p.416–423.
3. **Ford JR., Fulkerson L.R.** Flows in Networks, Princeton: Princeton University Press, 1962, p. 216.
4. **KhalifaA. F., Badr E.** Deep Learning for Image Segmentation: A Focus on Medical Imaging. Computers.Materials & Continua [online]. 2023.75, (1), p. 1995–2024.
5. **Zhang B., Wang J.** (ed.). Graph-Based Methods in Computer Vision: Developments and Applications. Hershey, PA: IGI Global. 2012. p.395.
6. **NHTSA.** Automated Vehicles for Safety. Available online: www.nhtsa.gov/technology-innovation/automated-vehicles-safety (accessed on 23 July 2023)
7. **Mathavan S.** A. Augmented Reality in Military Applications, International Journal of Engineering and Advanced Technologies, 9 (1), 2019, p. 51-54.

8. Yu Y., Yang B., Huang F., Fu Q., Kou R., Fu Q.,Mingliang G., Techniques and Challenges of Image Segmentation: A Review. Electronics [online], 2023, (12) 5, p. 1199.

9. **Ždímalová M., Boratková K.** Optimized GrabCut Algorithm in Medical Image Analyses.Lecture Notes in Networks and Systems, 2023, 608, p.101–116.

10. Kaplan, A., & Haenlein, M. (2019). Siri, Siri, in My Hand: Who's the Fairest in the Land? On the Interpretations, Illustrations, and Implications of Artificial Intelligence. Business Horizons, 62, 15-25. https://doi.org/10.1016/j.bushor.2018.08.004

11. Kaggle. (n.d.). *Datasets*. Retrieved July 9, 2025, from https://www.kaggle.com/datasets