Secure Systems Design and Technology Development. Veliko Tarnovo, Bulgaria
Proceedings of the International Scientific Conference, 2026. Volume III, 297-302

A Computer Vision Approach to Military Object Recognition

Krasimir Slavyanov
Department Computer Systems and Technologies
Vasil Levski National Military University
Shumen, Bulgaria
k.slavyanov@nvu.bg

***Abstract*—This research described is focused on computer vision tools implemented for detection of a particular military object (or many objects) in a observed scene, given a reference image(s) of the object(s). An algorithm for detecting a specific object based on finding point correspondences between the reference and the target image is designed. It can detect objects despite a scale change or in-plane rotation. It is also robust to small amount of out-of-plane rotation and occlusion. Reference images and a dataset with images obtained from a ground camera are used for the experiments. Detection of point features in images and extraction of feature points are designed for Localization of the recognized object(s) in a scene using matches of feature points (groups of points).**

Keywords — computer vision, military objects, reference images, speeded-up robust features

I. INTRODUCTION

The application of object detection and recognition provides the ability to recognize objects with their 2D positions in the image [1], [2]. Apart from machine learning approaches [3], deep learning [4] and neural network implementation [5], [6], this approach is solving the problem of how to detect a particular military object (or many objects) in a different operational environment, given a reference image(s) of the object(s) using only some common computer vision functions that can be found in the MATLAB. Local features and their descriptors, which are compact vector representations of the local environment, are the building blocks of many computer vision algorithms like Yolo for fast, accurate real-time object detection, segmentation, and classification [7], [8]. Their applications include image registration, object detection and classification, and motion tracking and estimation. The use of local features allows these algorithms to better cope with changes in scale, rotation, and occlusion. An algorithm for detecting a specific object based on finding point correspondences between the reference and the target image(photo) is presented. The implementation of the spatial orientation approach and the precise local positioning approach like in [9] can be implemented for calculation of the exact time for the photo capture. The object detection in photos works best for objects that exhibit non-repeating texture patterns, which give rise to unique feature matches. This technique is not likely to work well for uniformly colored objects, or for objects containing repeated patterns.

The algorithm is designed for detecting specific military object(s), for example, military vehicles. Detectors that rely on gradient and intensity variation approaches detect contrasting local features. These features include edges, spots, and regions. The selected local features show repeatable detections. Given two images of the same scene, most of the features that the detector finds in both images are the same. The features are robust to changes in viewing conditions and noise. Even confidential information can be hidden in an image using steganographic approach [10].

Effective military object detection can handle different tactical scenes: changes to the object size, location, orientation, and other challenges. Computer Vision System Toolbox in MATLAB offers a variety of techniques for handling challenges in military object detection.

This approach example shows how to detect a particular object in a panorama scene, given a reference image of the military object – a vehicle. The example presents an algorithm for detecting a specific military object based on finding point correspondences between the reference and the target image. It can detect objects despite a scale change or in-plane rotation. It is also robust to small amounts of out-of-plane rotation and occlusion.

II. MATERIALS AND METHODS

For better understanding the research is structured in several stages, each with specific purpose.

Online ISSN 3044-7771
https://doi.org/10.68302/std2026.vol3.154

A. Create a dataset with detailed reference images of specific military vehicles

To detect a specific military vehicle(s) a dataset of reference images should be created (or later updated).

Creating a set of reference images allows the detection of a change in location or reappearance of military objects of interest.

An important condition is that military objects - vehicles - are captured with higher image quality (brightness, sharpness, contrast etc.). For that stage of the approach three well know vehicles are captured to produce the dataset of referent image – Land Rover Defender, High Mobility Multipurpose Wheeled Vehicle (HMMWV or more popular as Humvee), named here as humvee1 and another version of the HMMWV – humvee2, all shown on fig. 1.

a) Land Rover Defender

b) HMMWV – humvee1

c) HMMWV – humvee2

Fig. 1. Referent images of a military vehicles: Land Rover Defender, HMMWV – humvee1, HMMWV – humvee2.

B. Observation and photo capturing.

The main goal at this stage is the creation of a dataset with images obtained from a ground camera, a drone camera, or images generated by artificial intelligence, in which one or more military vehicles are captured. A small set of complex images including more than one military vehicle is required.

The overall image can be with a scale change or in-plane rotation, with a small amount of out-of-plane rotation and occlusion. For that purpose, just one image out of the dataset is needed. One selection of all the mentioned before military vehicles is shown on fig.2.

Fig. 2. An image with some of the predefined military vehicles on the field - a target for the recognition process.

C. Read the reference image containing the object of interest

At this stage a reading of the target images is conducted in a MATLAB environment. Reading the reference image(s) containing the object of interest into MATLAB. The *rgb2gray* function is used for this purpose.

```
defenderImage =
imread('defender.jpg');
defenderImage =
rgb2gray(defenderImage);
figure;
imshow(defenderImage);
title('Image of Defender');
sceneImage = imread('vehicles.jpg');
sceneImage = rgb2gray(sceneImage);
figure;
imshow(sceneImage);
title('Image of a scene'
```

The images are converted from colored RGB type to greyscale images. The result for the Defender vehicle is shown on fig 3.

Fig. 3. A Greyscale reference image of Land Rover Defender

The greyscale image of the observation capture is on fig. 4.

Fig. 4. The observation capture in greyscale

D. Detect point features in both images

The function *detectSURFFeatures* for SURF (Speeded-Up Robust Features) method [11] is implemented, using only classical computer vision techniques, without neural networks. A variable is created with the result of the detection of characteristic points of the entire image (Defender).

```
defenderPoints =
detectSURFFeatures(defenderImage);
scenePoints =
detectSURFFeatures(sceneImage);
```

Plotting the most contrasting 100 points caried out with the function *selectStrongest*. Visualization of the strongest features of the reference image is on fig. 5.

```
figure;
imshow(defenderImage);
title('100 Strongest Point Features
from Defender Image');
hold on;
plot(selectStrongest(defenderPoints,
100));
```

Fig. 5. Visualization of the strongest features of the reference image

Visualizing the strongest features of the target image (fig. 6):

```
figure;
imshow(sceneImage);
title('300 Strongest Point
Features from scene Image');
hold on;
plot(selectStrongest(scenePoints,
300));
```

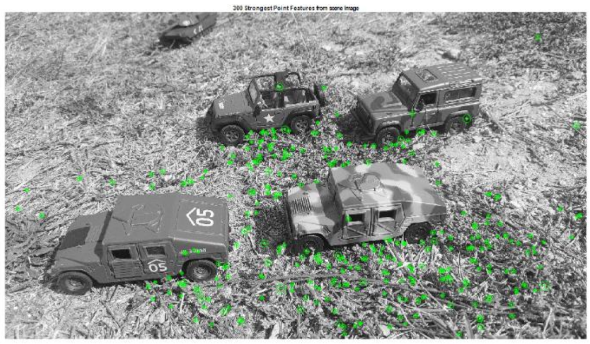

Fig. 6. Strongest features of the target image

E. Extract feature points (groups of points) and find extracted matching features

The extraction of the features of the points of interest in both images is executed with the functions *extractFeatures.* The function *matchFeatures* is used for finding the putatively matching features(pairs).

```
[defenderFeatures, defenderPoints] =
extractFeatures(defenderImage,
defenderPoints);
[sceneFeatures, scenePoints] =
extractFeatures(sceneImage,
scenePoints);
defenderPairs =
matchFeatures(defenderFeatures,
sceneFeatures);
```

The creation of an image of the matched points is depicted on fig. 7.

```
matcheddefenderPoints =
defenderPoints(defenderPairs(:, 1),
:);
matchedscenePoints =
scenePoints(defenderPairs(:, 2), :);
figure;
showMatchedFeatures(defenderImage,
sceneImage, matcheddefenderPoints,
matchedscenePoints, 'montage');
title('Putatively Matched Points
(Including Outliers)')
figure;
showMatchedFeatures(defenderImage,
sceneImage, matcheddefenderPoints,
matchedscenePoints, 'montage');
```

```
title('Putatively Matched Points
(Including Outliers)');
```

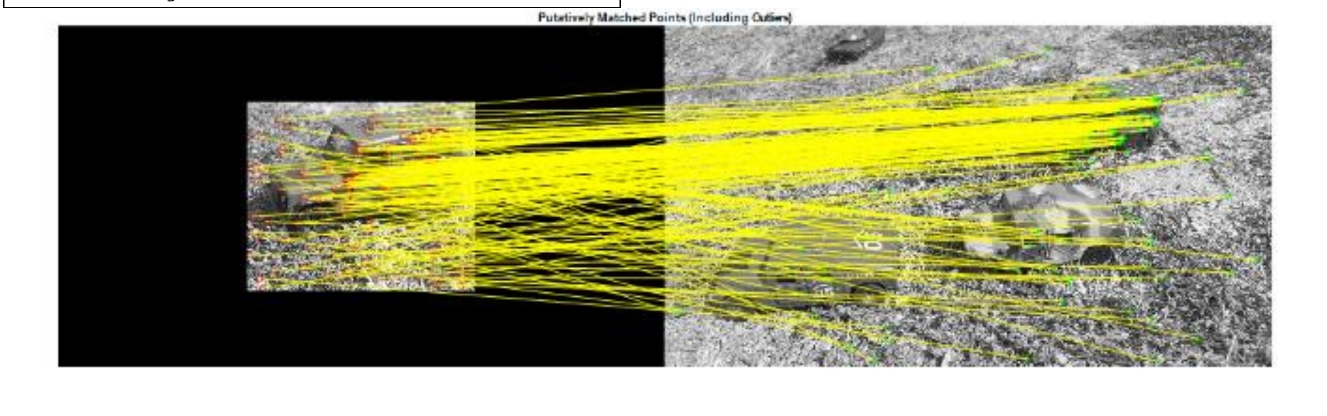

Fig. 7. Images of the matched points in the referent image and the captured image

F. Localization of the recognized object(s) in the scene using matches of feature points (groups of points).

At this stage a localization of the object in the scene is performed using presumed matches. The *estimateGeometricTransform* function is applied for the calculation of the transformation connecting the matching points while eliminating outliers. This transformation allows us to localize the object in the scene.

```
[tform, inlierdefenderPoints,
inlierscenePoints] = ...
estimateGeometricTransform(matchedde
fenderPoints, matchedscenePoints,...
'affine');
```

The matching pairs of points with outliers removed can then be displayed. The result of that procedure is depicted on fig. 8.

```
figure;
showMatchedFeatures(defenderImage,
sceneImage, inlierdefenderPoints,
inlierscenePoints, 'montage');
title('Matched Points (Inliers
Only)');
```

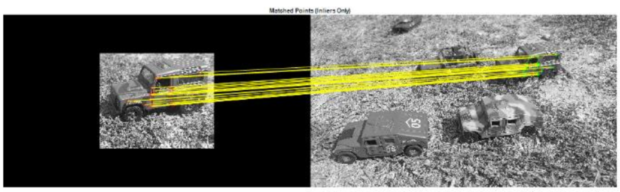

Fig. 8. The matching pairs of points with outliers removed for the recognized Land Rover Defender

A bounding polygon can then be formed around the recognized military vehicle:

```
defenderPolygon = [1, 1];...  % up
left
size(defenderImage, 2), 1;...  % up
right
size(defenderImage, 2),
size(defenderImage, 1);...  % down
right
1, size(defenderImage, 1);...  %
down left
1, 1];  % up left gain (closing)
```

To specify the location of the object in the scene, we can transform the polygon into the coordinate system of the target image (depicted on fig. 9):

```
newdefenderPolygon =
transformPointsForward(tform,
defenderPolygon);
% creating figure
figure;
imshow(sceneImage);
hold on;
line(newdefenderPolygon(:, 1),
newdefenderPolygon(:, 2), 'Color',
'y', 'LineWidth', 3);
title('Detected Defender');
```

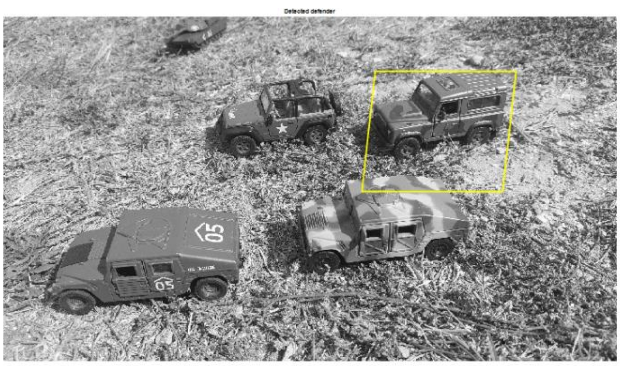

Fig. 9. Bounding polygon formed around the recognized military vehicle

III. RESULTS AND DISCUSSION

Similarly, after the evaluation of the same six stages, the other objects in this photo (humvee1 and humvee2) can be found, and they are finally marked with a different color. The results from the recognition of humvee1 and humvee2 are represented on fig. 10 and fig. 11.

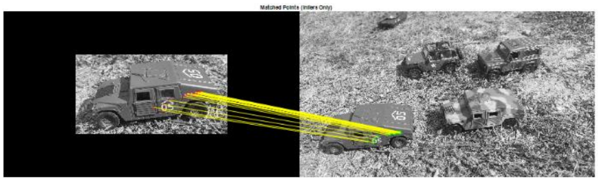

Fig. 10. The matching pairs of points with outliers removed for the recognized HUMVEE1

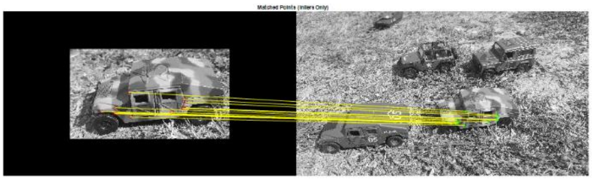

Fig. 11. The matching pairs of points with outliers removed for the recognized HUMVEE2

The three military vehicles can be recognized and then displayed on the target image with different colors – displayed on fig. 12.

```
figure;
imshow(sceneImage);
hold on;
line(newdefenderPolygon(:, 1),
newdefenderPolygon(:, 2), 'Color',
'y', 'LineWidth', 3);
line(newhumveePolygon(:, 1),
newhumveePolygon(:, 2), 'Color', 'b',
'LineWidth', 3);
line(newh2Polygon(:, 1),
newh2Polygon(:, 2), 'Color', 'r',
'LineWidth', 3);
title('Detected Defender, humvee 1
and humvee 2');
```

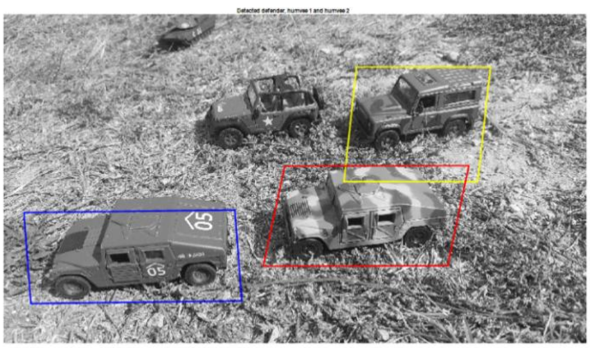

Fig. 12. The recognized Land Rover Defender, HUMVEE1 and HUMVEE2

IV. CONCLUSIONS

This approach of object detection works best for military objects that exhibit non-repeating texture patterns, which give rise to unique feature matches. This technique is not likely to work well for uniformly colored objects or for objects containing repeating patterns. Note that this algorithm is designed for detecting a military object, for example, well known military vehicles in reference images, rather than any vehicle. Computer vision recognition techniques can be used for clearly depicted and detailed objects. When recognizing partially hidden or significantly altered objects, better results could be obtained using artificial neural networks.

ACKNOWLEDGMENTS

This work was supported by the National Science Program "Security and Defence", which has received funding from the Ministry of Education and Science of the Republic of Bulgaria under the grant agreement no. Д01-74/19.05.2022. The material reflects only the author's opinion, and the Ministry of Education and Science is not responsible for the content.

REFERENCES

[1] H.Namdari, D.Nanda, X.Yuan, (2022). Progress in Computer Vision: Object Recognition. In: Albert, M.V., Lin, L., Spector, M.J., Dunn, L.S. (eds) Bridging Human Intelligence and Artificial Intelligence. Educational Communications and Technology: Issues and Innovations. Springer, Cham. https://doi.org/10.1007/978-3-030-84729-6_5

[2] M. Vashisht and B. Kumar, "A Survey Paper on Object Detection Methods in Image Processing," 2020 International Conference on Computer Science, Engineering and Applications (ICCSEA), Gunupur, India, 2020, pp. 1-4, https://doi.org/10.1109/ICCSEA49143.2020.9132871.

[3] T. A. Nayeem, S. M. Motaharuzzaman, A. T. Hoque and M. H. Rahman, "Computer Vision Based Object Detection and Recognition System for Image Searching," 2022 12th International Conference on Electrical and Computer Engineering (ICECE), Dhaka, Bangladesh, 2022, pp. 148-151, https://doi.org/10.1109/ICECE57408.2022.10089019

[4] I. Atik, "Deep Learning In Military Object Detection: An Example of The Yolo-Nas Model," 2024 8th International Symposium on Innovative Approaches in Smart Technologies (ISAS), İstanbul, Turkiye, 2024, pp. 1-7, https://doi.org/10.1109/ISAS64331.2024.10845459.

[5] J. Cherapanamjeri and B. N. K. Rao, "Neural Networks based Object Detection Techniques in Computer Vision," 2022 4th International Conference on Inventive Research in Computing Applications (ICIRCA), Coimbatore, India, 2022, pp. 1092-1099, https://doi.org/10.1109/ICIRCA54612.2022.9985581.

[6] S. Kaushik, A. Raman and K. V. S. Rajeswara Rao, "Leveraging Computer Vision for Emergency Vehicle Detection-Implementation and Analysis," 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Kharagpur, India, 2020, pp. 1-6, https://doi.org/10.1109/ICCCNT49239.2020.9225331

[7] S. Singh and R. G N, "Military Based Object Detection in Satellite Imagery by Optimising YOLOv8," 2024 IEEE Space, Aerospace and Defence Conference (SPACE), Bangalore, India, 2024, pp. 165-168, https://doi.org/10.1109/SPACE63117.2024.10667819

[8] S. Ali, Abdullah, A. Athar, M. Ali, A. Hussain and H. -C. Kim, "Computer Vision-Based Military Tank Recognition Using Object Detection Technique: An application of the YOLO Framework," 2023 1st International Conference on Advanced Innovations in

Smart Cities (ICAISC), Jeddah, Saudi Arabia, 2023, pp. 1-6, https://doi.org/10.1109/ICAISC56366.2023.10085552

[9] V. Atanasov, T. Trifonov, "A new model of ultra wideband sensors based interactive system", International Journal on Information Technologies and Security, № 1 (vol. 16), 2024, pp.39-48, ISSN 1313-8251, https://doi.org/10.59035/XYCG3094

[10] V. Stoyanova, I.Lilov "Steganographic algorithm using the different color components of a 24-bit image", Security & Future, WEB ISSN 2535-082X; PRINT ISSN 2535-0668, Vol. 5 (2021), Issue 2, pg(s) 61-63

[11] https://www.mathworks.com/help/matlab/index.html [Accessed: Apr. 05, 2026].