Automatika

Journal for Control, Measurement, Electronics, Computing and Communications

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/taut20

Recognition of dynamic objects from UGVs using Interconnected Neural network-based Computer Vision system

Barış Gökçe & Güray Sonugür

To cite this article: Barış Gökçe & Güray Sonugür (2022) Recognition of dynamic objects from UGVs using Interconnected Neural network-based Computer Vision system, Automatika, 63:2, 244-258, DOI: 10.1080/00051144.2022.2031539

To link to this article: https://doi.org/10.1080/00051144.2022.2031539

Published online: 28 Jan 2022.

AUTOMATIKA
2022, VOL. 63, NO. 2, 244–258
https://doi.org/10.1080/00051144.2022.2031539

Recognition of dynamic objects from UGVs using Interconnected Neural network-based Computer Vision system

Barış Gökçe [a] and Güray Sonugür [b]

[a]Department of Mechatronic Engineering, Necmettin Erbakan University, Konya, Turkey; [b]Department of Mechatronic Engineering, Afyon Kocatepe University, Afyonkarahisar, Turkey,

ABSTRACT

In this study, moving object recognition is performed by using images from a camera mounted on an unmanned ground vehicle. A GPS coordinate-based algorithm has been developed to obtain moving object silhouettes. In order to classify these silhouettes, an interconnected artificial neural network (ICANN) architecture consisting of two stages has been developed. The method consists of two phases. In the first phase, real-time images are converted to binary images at the end of the GPS-assisted image registration process. Then, the silhouettes are extracted from the background of the images using connected component labelling. In the second phase, two interconnected neural networks are used. The first neural network classifies silhouettes as objects or noise. The second neural network divides objects into seven subclasses as pedestrians, potholes, cars, etc. Compared to CNN-based techniques, a simpler NN architecture was employed in the research, and better accuracy rates were achieved with fewer samples. Another contribution of the research is simultaneous localization and mapping (SLAM) applications can be performed in non-GPS environments using pre-recorded images containing GPS information. In experimental studies, maximum success rates of 96,1% in object classification were obtained. The results obtained were compared to YOLO, the recently popular algorithm for object recognition.

ARTICLE HISTORY
Received 8 July 2021
Accepted 14 January 2022

KEYWORDS
Moving object recognition; GPS assisted recognition; Robot vision; neural networks; image registration

1. Introduction

Object detection is the first stage of object recognition studies and has indispensable importance for intelligent robotics and autonomous systems [1]. These techniques are used to understand and analyze any scene in images or videos. Its main application areas include target recognition and tracking [2], face detection, optical character recognition, agricultural disease recognition, and pedestrian detection for driving assistance systems [3–5]. Nowadays, intensive studies on unmanned vehicles have led to increased research on moving object recognition (MOR). Especially, studies on unmanned ground vehicles (UGV) [6,7], unmanned aerial vehicles (UAV) [8,9], advanced driving assistance systems [10], and unmanned submarine vehicles [11] have increased. The studies on MOR in the literature can be grouped under two main headings. Studies with fixed cameras [12,13] and moving cameras [9,14]. In the case of fixed cameras, the background is fixed. All pixels showing a significant change between two consecutive frames belong to moving objects except for noises. Consequently, the performance of moving object detection is high, and its computational costs are low. The work done with moving cameras can also be examined in two categories: video streaming (offline) and real-time images (online). The background images change due to the camera's movement when moving cameras are used. The algorithms to be used have to distinguish the images, which seem like moving because of the camera's movement and the real moving objects. There is no time limitation in offline studies [15]. There is sufficient time for the revision of experimental studies and methods. It is used to identify important patterns such as crime detection and suspect behaviour detection from large amounts of video data [16]. Real-time work is much more challenging. There is limited time to detect foreground and background changes, detect objects and recognize them. Otherwise, the installed real-time system will fail and will not achieve its purpose. Another constraint is computational cost and processing time. In real-time moving object detection and recognition, it is necessary to revise the methods to decrease the computational costs, increase the processing speed and regulate some conditions in advance. Therefore, pre-processing is generally required [7,17,18]. Hussain et al. [19] developed an object recognition framework based on whale optimization to address all of these challenges. The first step in MOR is to detect objects. In the moving object detection (MOD) step, the foreground and background

CONTACT Güray Sonugür gsonugur@aku.edu.tr Department of Mechatronic Engineering, Afyon Kocatepe University, Afyonkarahisar, 03200, Turkey

are separated. Moving objects are usually obtained as binary large objects (BLOBs). The methods used for MOD are evaluated under three main steps: modelling-based background subtraction, optical flow, and frame differencing. Modelling-based background subtraction is a widely used method in MOD. In this method, changes in the pixel values of the images are observed in video streams. These changes are modelled with statistical methods, and each pixel is classified as a background or foreground. Single Gaussian [20], Mixture of Gaussian (MOG) [21], adaptive Gaussian [22], and Spatio-temporal [23] methods are used in the modelling. If all pixels are modelled, the processing time on the computer is quite long. For this reason, descriptors or features are often used instead of all pixels. Moving objects can be detected with these methods, but they are susceptible to noise [24]. The optical flow methods [25] give stable results in detecting objects from the moving cameras. These methods can be applied as feature-based or area-based [26]. However, due to the low time interval between the successive frames and the movement of the background, the computational cost of optical flow methods is high. As a result, unless you have specialized hardware, using these approaches is inconvenient [27]. The frame differencing method [28] is an easy-to-understand and low-cost method. It is based on taking the differences between consecutive frames. However, it fails to detect the entire object [24]. MOR involves more complex processes than MOD. Therefore, fewer studies are encountered in the literature. In general, the studies are performed on image data sets [29] or video data sets [15]. Heuristic methods are generally used as recognition methods due to successful results. CNN, one of the most popular methods in recent years, is one of these methods [30]. CNN is a technique that is becoming more popular in a variety of fields. It is utilized in topics like facial recognition[31], network security [32], and text recognition [33]. Some of the most common CNN-based approaches today are Region-based CNN (R-CNN) [34], Fast R-CNN [35], Faster R-CNN [36], and You Only Look Once (YOLO) [37]. Among these approaches, YOLO is the most effective for real-time object recognition [38]. It is widely used to recognize traffic signs encountered by autonomous vehicles during their navigation [39,40]. In addition, object recognition is performed with modified versions [41]. Although a high proportion of background modelling is used to obtain blobs representing moving objects, very few GPS-based techniques have also been used. In a study by Kong et al. [42], abandoned objects were detected using video frames. For this purpose, the videos retrieved from the same GPS coordinates were roughly aligned and compared. In the study, no results indicating accuracy or precision have been published. The success rate was low due to the rough matching process. The objects in the scene were detected in the study by Morales et al. [7], by comparing object-free and object-containing video frames retrieved from the same GPS coordinates. Accuracy values were not given in the paper. Only computational times were published. Accordingly, the computational times vary between 20 and 200 ms depending on the object density. Furthermore, because the images were recorded monthly, daily, and hourly due to the used algorithm, specific equipment capable of storing large amounts of data was necessary.

In this study, real-time object recognition was performed using images obtained from a camera-mounted unmanned ground vehicle. MOR studies performed on unmanned vehicles must be in real-time. Therefore, a GPS-assisted algorithm was developed for the required high processing speed. A two-stage interconnected artificial neural network (ICANN) was established for MOR processes. In the first stage; Instead of the noise filtering process of the images, a neural network (NN) was created that can learn the difference between noise and the object. In the second stage, BLOBs selected as objects are classified as pedestrians, group of pedestrians (GoP), cars, riders, potholes, speed bumps, and dogs. One of the main things for a UGV is recognizing objects that are likely to come across on the road. For this reason, seven objects are specified for classification. They are objects where a UGV would most likely meet on the road, and when that occurs, the UGV needs to respond. For example, if there is a pothole in front of the UGV, it must slow down or manoeuver. Thus, no time was lost in recognizing unnecessary objects to take action for UGV.

The main contributions of this paper are as follows:

- Thanks to the object-free images taken from preselected GPS coordinates, a constant background for each scene was obtained during the real-time movement. Thus, the difficulties caused by the background's irregular changes have been overcome.
- In comparison to CNN-based techniques, a simpler NN architecture was employed in the research, and better accuracy rates were achieved with fewer samples. As a result, the computational time was decreased even further, and the time constraint in real-time motion was overcome.
- Even if there are no GPS signals on the studied routes, simultaneous localization and mapping (SLAM) applications can be performed thanks to pre-recorded image sets, including GPS information.
- All object-free background images on the routes are recorded based on geographical coordinates. Thus, modifications or anomalies that may occur in the scenes over time can be easily detected. Moreover, periodic changes can be detected in such areas.

The rest of this paper is organized as follows. In section 2, the whole algorithm is explained. Pre-processes are presented in detail in Section 3, and real-time operations in Section 4. Results and conclusions are presented in Section 5 and Section 6, respectively.

2. Suggested approach

Moving object recognition was performed in two phases. The first phase is the pre-processing phase, and the second phase is the real-time phase. The real-time processing phase is also divided into MOD and MOR sub-phases. Preliminary preparations are required along the path of the UGV to increase the object detection speed. For this purpose, some GPS coordinates are selected as the reference and symbolized K_1,K_2,... K_n at short intervals along the route. The coordinate information includes latitude, longitude, and height information. Images are taken by a camera when there are no moving objects in the scene at each reference coordinate. These images are represented by symbols $\{RI_1, RI_2, \ldots RI_n\}$ and are referred to as reference images. The reference images and coordinates are demonstrated in Figure 1.

The reference coordinates, the reference images, and the lighting values of the scenes are recorded in the relational database. In the real-time processing phase, the UGV takes a snapshot from the reference coordinates and compares it to the reference image of the same coordinate in the memory. The two images are geometrically aligned before comparison because of angular and linear differences between the images. The comparison process is done between binary images. If the resulting blobs are classified as noise in the MOR sub-phase, the UGV continues moving. If classified as an object, the process continues according to the object class.

3. Pre-processing

3.1. Creation of image and GPS coordinate database

Javad Triumph-1M model GPS device is used for reference coordinate determination. The horizontal and vertical sensitivities of this device are below 1 cm. The refresh frequency of the device is 10 Hz. The operations performed in this section are part of the preparation phase. The transactions are as follows.

(1) The reference coordinates are taken at approximately 1 m intervals. The UGV is able to update GPS coordinates between reference coordinates at both speeds of 3 and 4 km/h. (2) When there are no moving objects in the scene, images are taken from the reference coordinates, and their features are extracted. Thus, by saving the feature vectors to the database during the pre-processing phase, no time is spent extracting the features of the reference image in the real-time phase. Feature extraction methods will be discussed in the "Image Registration" section. (3) The UGV has a sensor that measures the brightness of daylight. Differences due to weather conditions between reference and snapshot pairs are eliminated using measured values from the sensor. (4) The data mentioned above (reference coordinates, reference images, features, and brightness values) are specific to each reference coordinate. They are associated with each other and saved in the database.

3.2. Creation of image dataset

The effect of the rolling shutter should be taken into account in the image acquisition studies with cameras mounted on moving platforms. In order to achieve accurate results, this effect must be compensated [43].

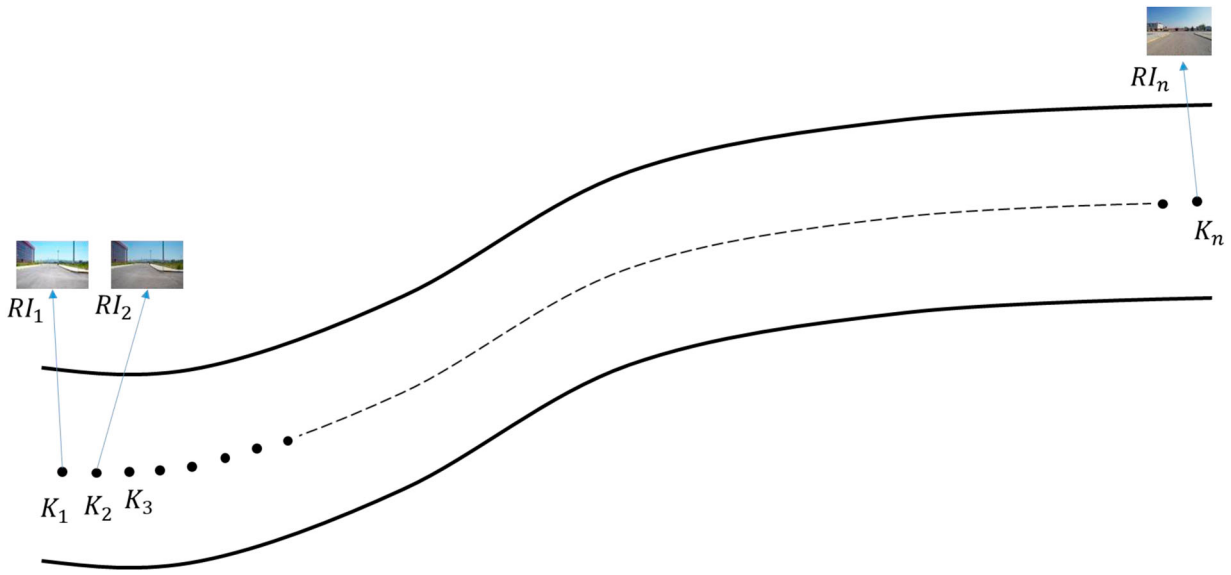

Figure 1. Reference coordinates along the route of the UGV and reference images taken from these coordinates.

In order to suppress this effect, a lowpass filter was applied to the images [44].

ICANN is composed of NNs performing two different tasks. These NNs are designed according to the principle of supervised learning. The better the inputs are selected and organized in supervised learning, the better the relationship to the outputs is expressed. Therefore, it is necessary to collect sample images to be used to train NNs. We took sample images on the route where the real-time movement will be performed. We have endeavoured to let NN learn as many blobs as possible by taking images from different camera viewpoints in different positions and locations.

3.3. Data pre-processing

All the images were converted into binary images by morphological operations. After the conversion, all the blobs obtained by Connected Component Labelling (CCL) method were extracted. If the blob represents an object, it is labelled according to its class. Otherwise, it is labelled as noise. Some samples of object and noise blobs are shown in Figure 2.

In this way, the image dataset to be used in training was made ready. In the next step, two types of features group are serially fused. The first feature group consists of 6 standard features: area, perimeter, solidity, etc. The other group contains 72 features that represent the distances and angles of boundary lines at 10-degree intervals. Fusing features have also been used in different studies. One of them is the machine inspection application by Hussain et al. [45]. Figure 3 illustrates how to obtain these features.

In Figure 3, $(d_1 \ldots d_{36})$ represents the amplitude of the vectors between the border coordinates, which are arranged in 10^o intervals, $(\theta_1 \ldots \theta_{36})$ represents

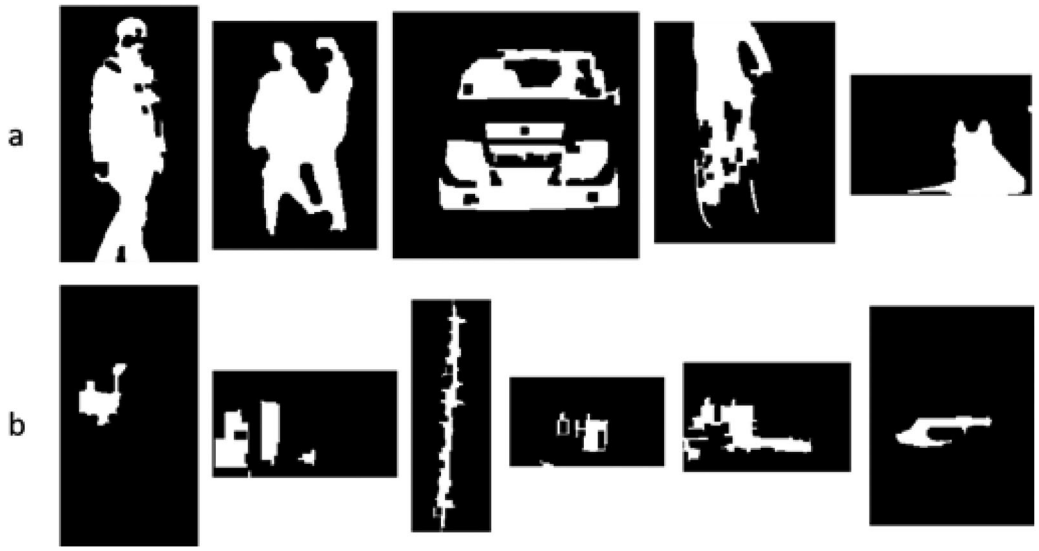

Figure 2. (a) Samples of object blobs, (b) Samples of noise blobs.

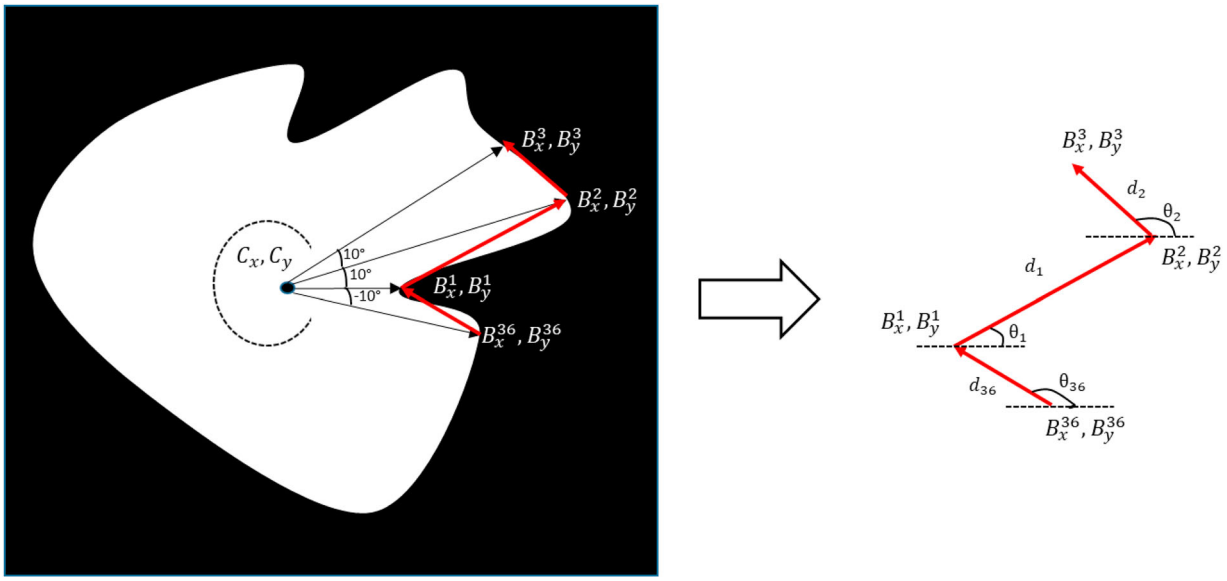

Figure 3. A formal expression of BLOB's feature extraction.

the angles of these vectors, (C_x, C_y) denotes the centre of gravity of the blob and $(B_x^1, B_y^1) \ldots (B_x^{36}, B_y^{36})$ indicates the coordinates of the border points. The second portion of the fused feature is composed of d_i and θ_i $(i = 1to36)$. They are calculated as shown in Equation 1, 2.

$$d_i = \sqrt[2]{(B_y^{i+1} - B_y^i)^2 + (B_x^{i+1} - B_x^i)^2} \quad (1)$$

$$\theta_i = \tan^{-1} \frac{B_y^{i+1} - B_y^i}{B_x^{i+1} - B_x^i} \quad (2)$$

3.4. Training of the icann

ICANN includes three training processes. These processes were; training of object/noise classifier and training of object type classifier. The schematic representation of each process is given in Figure 4 a, b, and c. A dataset of 1325 blobs was created for the training. Seven hundred eighty of these blobs were used for object/noise classification and 830 for object classification. 80% of this dataset was used for training, 10% for testing, and the remaining 10% for verification. Seventy-eight formal features were extracted from each blob.

Levenberg-Marquardt [46,47] and Scaled Conjugate Gradient [48] methods were used as the backpropagation algorithm. The Levenberg-Marquardt method gives much better results in terms of both performance and processing speed. The Levenberg-Marquardt algorithm was selected because the processing speeds of computers are significant in real-time motion.

Several experimental tests were performed to build the ICANN architecture that provides high efficiency and low processing time. Table 1 shows the results for the four better architectures in classifying object/noise and object type.

In Table 1, HLS represents the hidden layer size, TP Rate represents the true positive rate in the confusion matrix obtained after training, and computational time is the average recognition time per sample. The performance metric used in the analysis was calculated as shown in Equation 3. The Model-3 architecture, which

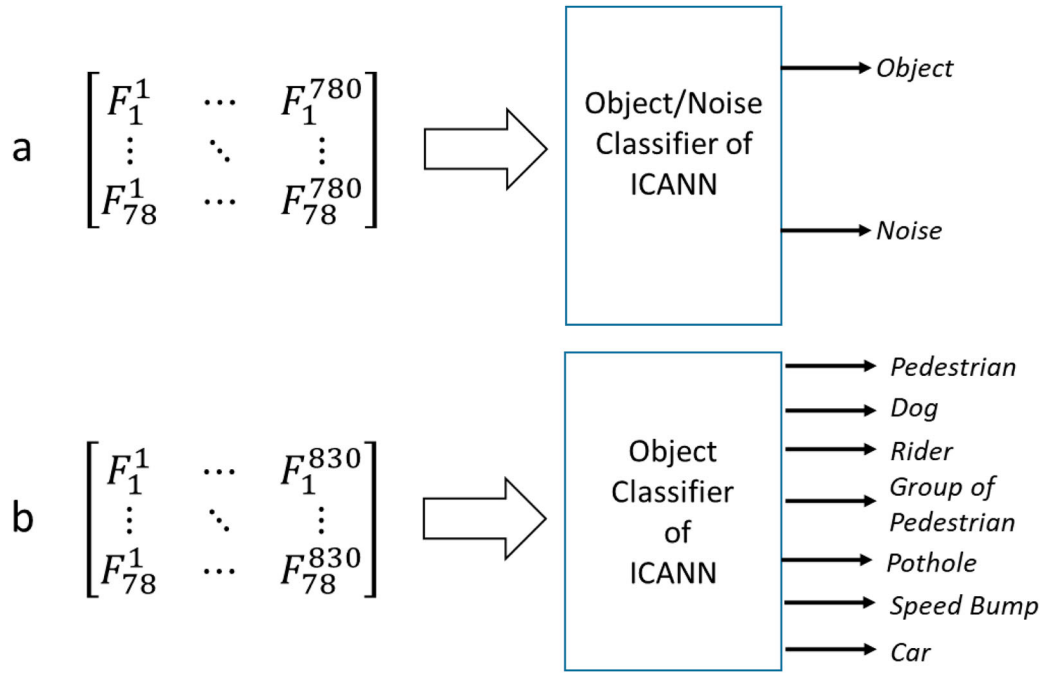

Figure 4. The schematic representation of the ICANN process (a) object/noise classifier and (b) object type classifier.

Table 1. Results obtained in the ICANN training and processing time test.

	Model	HLS/Neuron Number	Epoch Number	TP Rate (%)	Performance (min is the Best)	Computational Time per Sample (ms)
Object/ Noise Classification	Model 1	1/[20]	21/200	95.4	13.64	16.9
	Model 2	2/[10 10]	16/200	97.1	12.26	17.6
	Model 3	2/[20 20]	39/200	98.5	7.37	17.9
	Model 4	3/[10 10]	92/200	98.5	12.37	18.5
Object Classification	Model 1	1/[20]	24/200	94.4	26.13	15.6
	Model 2	2/[10 10]	22/200	95.0	32.02	15.0
	Model 3	2/[20 20]	39/200	95.2	9.89	16.7
	Model 4	3/[10 10]	72/200	96.4	23.55	19.2

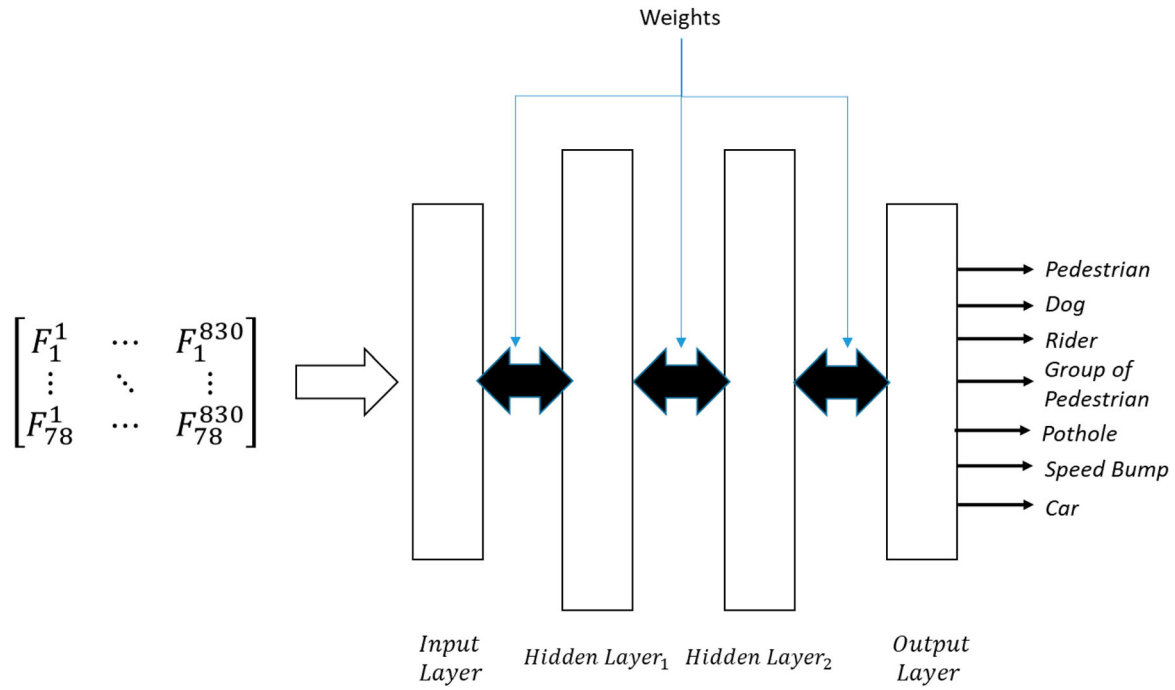

Figure 5. Object classifier architecture of ICANN.

provides the most accurate results for both classifications, was selected.

$$Performance = |Output_{desired} - Output_{calculated}| \quad (3)$$

Model-3 architecture is shown in Figure 5.

4. Real-time operations

In real-time motion, some processes are run sequentially. With the MOD process, moving objects are detected as blobs on the image. The MOR process starts immediately after this process. In this process, the real object and noise blobs detected in the previous step are first classified as objects or noise, and then the object recognition process is made by determining the classes of the objects. The UGV used in the study, shown in Figure 6, has two 420W DC motors and two BTS 7960 PN motor drivers. 1024 ppr encoder mounted on both wheels.

It has an NVIDIA Jetson TX2 development board and a ZED camera. Image processing was performed with 1280 × 768 resolution images using Open CV. The UGV operates at two speeds: 3 and 4 km/h. There are two batteries with 12 V 60A capacity providing power to the vehicle. Since we are using Jetson TX2 and Zed cameras, we have some benefits regarding disk capability. Jetson TX2 has an internal hard disk of 32 GB. It also supports up to 128 GB of SD Disk. Zed stereo camera

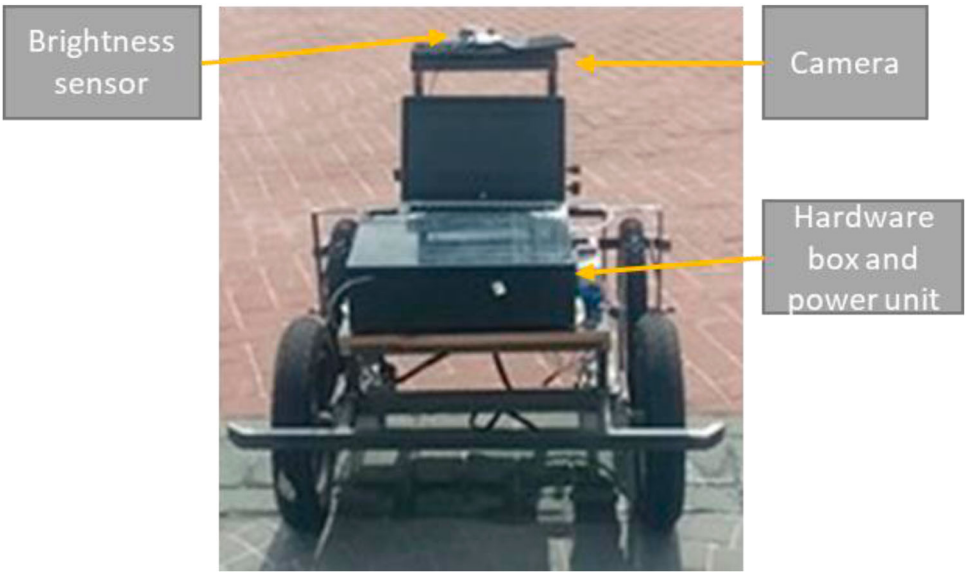

Figure 6. Rear view of the UGV used in experimental studies.

compresses the image frames and transfers them to disk in H264 format. Due to these hardware features, the disk requirement has decreased.

4.1. Image transformation

When UGV moves in real-time, it takes snapshots called $\{CI_1, CI_2, \ldots CI_n\}$ from each reference coordinate. The aim is to detect moving objects in the snapshot by comparing these images with reference images $\{RI_1, RI_2, \ldots RI_n\}$. For this purpose, each CI_i image is compared with RI_i of the same coordinate in the database at the snapshot time. However, due to the camera's movement (non-stationary camera), reference and snapshot pairs may be taken from different viewpoints and spatial points. Therefore, it is necessary to align the snapshots with the reference images. Image registration techniques are generally used for alignment. This technique consists of feature extraction, feature mapping, determination of the transformation method, and image recovery processes [49]. In this study, registration is performed using invariant features such as SURF [50], Harris [51], and Fast [52] algorithms. The number of features of the images differs according to the background complexity and methods. In the later step, many features are needed for optimal matching between snapshots and reference images. In real-time operation, the processing time of the feature extraction process is also essential. Therefore, an experimental study was performed to discover the method that extracts the most features in the shortest time. Results are given in Table 2.

Since the most appropriate values in processing time and the number of features were taken in the SURF method, SURF was chosen as the feature extraction method. In order to ensure sufficient processing time, only the geometric transformation method was used [53]. Euclidean distance (ED), Manhattan distance (MD), and Normalized Cross-Correlation (NCC) techniques were tested to calculate the shifting distance and angle between reference images and snapshots. The two images were geometrically aligned using these shifts, and the difference between them was detected. Difference pixels were regarded as they were moving objects. ED_k, $MD_k$and NCC_k vectors were calculated for feature matching as follows:

Table 2. The average number of features and computational times obtained as a result of the experimental studies.

	Noise		Object	
	Computational Time (s)	Number of Features	Computational Time (s)	Number of Features
FAST	0.16	45.11	0.13	635.1
HARRIS	1.31	3306.65	1.45	4596.24
SURF	0.76	304.14	0.76	1134.95

Table 3. Results obtained in the feature matching test.

	Length of Feature Vector	Number of Experiments	Computational time (sn)	Successfully Detection Rate
ED	64	102	0.068	%89
MD	64	102	0.054	%93
NCC	64	102	0.82	%74

Euclidean distance:

$$ED_k = \sum_{k=1}^{m}\sum_{j=1}^{n}\sum_{i=1}^{t}((FR_i^k - FI_i^j)^2)^{1/2} \quad (4)$$

Manhattan Distance:

$$MD_k = \sum_{k=1}^{m}\sum_{j=1}^{n}\sum_{i=1}^{t}|FR_i^k - FI_i^j| \quad (5)$$

NCC:

$$NCC_k = \frac{\sum_{k=1}^{m}\sum_{j=1}^{n}\sum_{i=1}^{t}(FR_i^k - FR_{avg}^k) * (FI_i^J - FR_{avg}^J)}{[\sum_{k=1}^{m}\sum_{j=1}^{n}\sum_{i=1}^{t}(FR_i^k - FR_{avg}^k)^2]^{1/2} * (FI_i^J - FR_{avg}^J)^2} \quad (6)$$

Here, *FR*; the feature vector of the reference image, *FI*; the feature vector of the snapshot, *t*; element numbers of feature vectors, *n*; the number of features of the snapshot, *m*; the number of features of the reference image, the arithmetic mean of the *avg* vector, and *i*, *j*, *k* are the index numbers of the feature vectors. The results from the experimental studies to determine the efficient method in feature matching are given in Table 3. As seen in Table 3, the NCC method was eliminated due to very high computational time and low success rate. The MD method was used in real-time studies due to its high success rate and low processing time.

After this process, the snapshot (CI_i) was aligned to the reference image RI_i. Image differencing method [54] and image binarization were used to detect moving objects present in the snapshot. In this method, it is essential to determine the threshold value for binarization. The objects will not fully appear or will be difficult to detect because the noise will increase when the incorrect threshold value is used. In order to prevent this fault, the illumination value of the scene was also recorded in the database. At the end of this stage, blobs are obtained. Some of the blobs are real objects, while others are noise. The next step is the MOR, in which the recognition processes are performed.

4.2. Mor

4.2.1. Creating interconnected neural network

The ICANN architecture created is shown in Figure 7. In this section, two interconnected NN are used. The first block is a classifier. The input vector consists of formal features of the blobs obtained in the MOD phase.

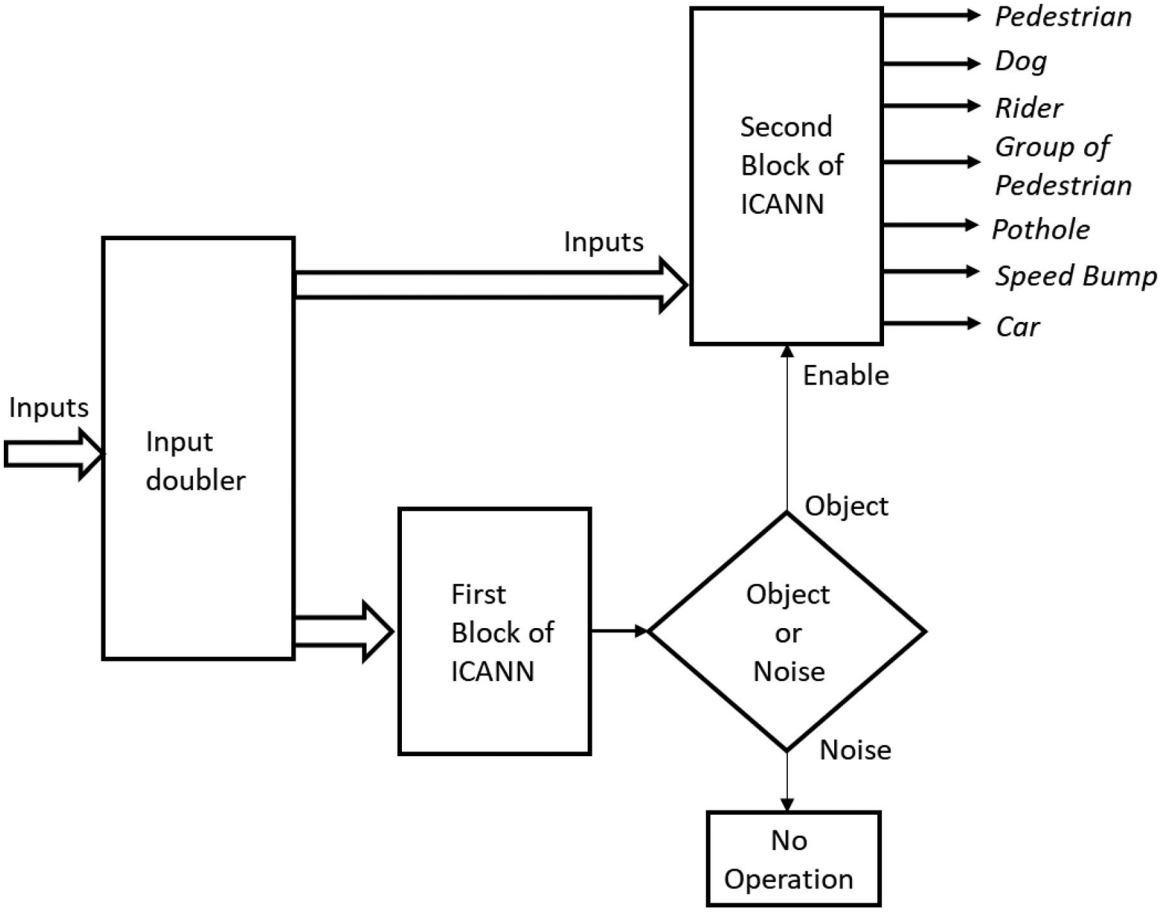

Figure 7. Complete architecture of ICANN.

At its output, it decides whether these inputs represent a real object or a noise. This block has a large share of the high success rate in the study. If trained strongly, it can achieve high successful classification rates. Therefore, long-lasting morphological processes are not required. The second block of ICANN is a classifier. Only the outputs classified as objects in the first block can be the inputs of these two blocks. The output layer of the second block has seven classes. These classes are selected from objects that are likely to encounter a UGV on the road during real-time movement. These are pedestrians, a group of pedestrians, bicycles, cars, potholes, speed bumps, and dogs.

5. Experimental results

The images used in the ICANN training process were taken from four separate places inside the campus. In addition, two different locations were used in real-time experiments. Images from the internet were also used in a few experiments. Any photos of the locations used are seen in Figure 8.

The routes within the campus used to generate the dataset and the number of test drives performed on each route is given in Table 4.

5.1. Training process

The success of the object/noise classification has a high share in the total success. The confusion matrix resulting from the training is given in Table 5.

In Table 5, the precision column shows the percentages of correct and incorrect predictions for each class, respectively. Diagonal cells correspond to correct predictions, non-diagonal cells correspond to incorrect predictions. The rows at the bottom of the table show the percentages of all samples for each class classified correctly and incorrectly.

In ICANN training, two datasets were used. The first dataset was used to classify object/noise, and the second for the object type classification. These datasets were obtained by performing a total of 11 test drives on four separate campus roads. When a test drive was made on all roads, 805 images were recorded. Object and noise features derived from these images were used for ICANN training. The first dataset consists of 960 blob samples, 560 objects, and 400 noise.

Each sample is represented in this dataset by a vector consisting of 78 features. 80% of this dataset was used for training, 10% for testing, and the remaining 10% for verification. Verification data was used for early stopping. Levenberg-Marquardt and Scaled Conjugate Gradient methods were used as the backpropagation

Figure 8. Photos from a few training and test locations.

Table 4. Routes used to create the image dataset.

Route Number	Route Length (m)	Effort	Number of laps
1	210	Training test	3
2	120	Training test	3
3	305	Training test	2
4	170	Training test	3
5	180	Real-time test	–
6	212	Real-time test	–

Table 5. Confusion matrix of Object / Noise classification.

	Classes	Object	Noise	Precision
Scaled conjugate gradient	Object	382	15	%96.22
		%48.9	%1.9	%3.78
	Noise	18	365	%95.3
		%2.3	%46.8	%4.7
	TP Rate	%95.5	%96.05	%95.76
		%4.5	%3.95	
Levenberg Marquardt	Object	395	7	%98.25
		%50.64	%0.9	%1.75
	Noise	5	373	%98.7
		%0.6	%47.8	%1.3
	TP Rate	%98.75	%98.16	%98.5
		%1.25	%1.84	

algorithm. As can be seen from Table 5, the Levenberg-Marquardt method gives much better results.

Features extracted from the 830 object blob were used in the object type classification process. Object classification is the second step of ICANN. Only the samples classified as objects in the first step can pass to this step. More sample blobs were used in this step to increase classification success 20% of the dataset was used for testing and validation, and 80% for training. Levenberg- Marquardt and Scaled Conjugate Gradient methods were used for training.

The results were obtained by the Levenberg -Marquardt method. There was no significant difference between the two methods in terms of performance and processing speed. According to the results, the accuracy rate of the classification performance was 95.22%. The best performances took place in the pedestrian, dog, and car classes.

5.2. Successful classification rates

In experimental studies, UGV was moved at two different speeds, 3 and 4 km/h. Real-time movements were performed on four different routes and in various weather conditions, and a total of 392 reference coordinates were passed. Some results from recognition tests with single objects are given in Figure 9.

Several criteria have been identified as performance metrics. The two main ones are single and multiple object recognition rates. In the scenes preparation for the test studies, UGV was enabled to come across obstacles such as pedestrians, cars, dogs, etc., while moving at the determined speed, and recognition performances were calculated. Real-time motion at 4 km/h is more challenging. Because the system has less time to perform MOD and MOR processes, it will be more affected by deviations due to environmental conditions. Table 6 shows the results obtained from the experimental studies at both speeds and where single objects were detected.

Table 7 shows the results obtained from the experimental studies in which multiple objects were detected at both speeds.

As can be seen from Table 7, the accuracy rates are quite high. The high accuracy rate shows the success of the object/noise classification performed in the first step. The developed approach is capable of recognizing all objects in the scene. Although there were small changes according to environmental conditions, no difference was observed in the studies at general at 3 and 4 km/h. The distance between two reference coordinates was determined as 1 m on the four routes where

Figure 9. (a) Original images used in single object recognition, (b) object recognition results.

Table 6. Successful classification rates in single object recognition tests at two different speeds.

	Accuracy for 3 km/h (%)	Accuracy for 4 km/h (%)
Pedestrian	92.2	90.1
GoP	90.4	87.2
Speed bump	96.1	92.3
Car	95.3	90.8
Rider	92.7	89.1
Pothole	81.6	81.6
Dog	90	87.3

Table 7. Successful classification rates in multiple object recognition tests at two different speeds.

	Accuracy for 3 km/h (%)	Accuracy for 4 km/h (%)
Pedestrian	87.4	82.6
GoP	83.1	80.4
Speed bump	93.5	91.6
Car	90.4	83.3
Rider	83.8	80.2
Pothole	80.6	79.2
Dog	86.8	81.2

the experimental studies were performed. According to this, the time interval between two reference coordinates at 3 km/h is 1.2 sec and at 4 km/h is 0.9 sec. The MOD and MOR processes must be completed within these periods. In real-time experimental studies, it was observed that the calculation time was not sufficient when the speed of UGV was higher than 5 km/h. Therefore, higher-level computer hardware should be used when UGV is operated at these speeds. Multiple objects recognized by ICANN are given in Figure 10.

5.3. Comparison with YOLO v3

The results obtained using ICANN were compared with those obtained with YOLO v3, one of the most popular object recognition methods of recent years, by conducting experimental studies. The YOLO v3 network was trained with 80 images per class under the same conditions as ICANN. The object recognition process was performed with YOLO v3 and ICANN in 140 images (30 pedestrians, 20 GoP, 20 cars, 20 bicycles, 20 dogs, 15 potholes, and 15 speed bumps). Accuracy and average computational times are given in Table 8.

After training under the same conditions as ICANN, YOLO v3 had lower image recognition performance. In order to make successful recognition with YOLO v3, it is necessary to use datasets consisting of thousands of images for each class in training. Some examples of images used for the recognition of objects by both methods are shown in Figure 11.

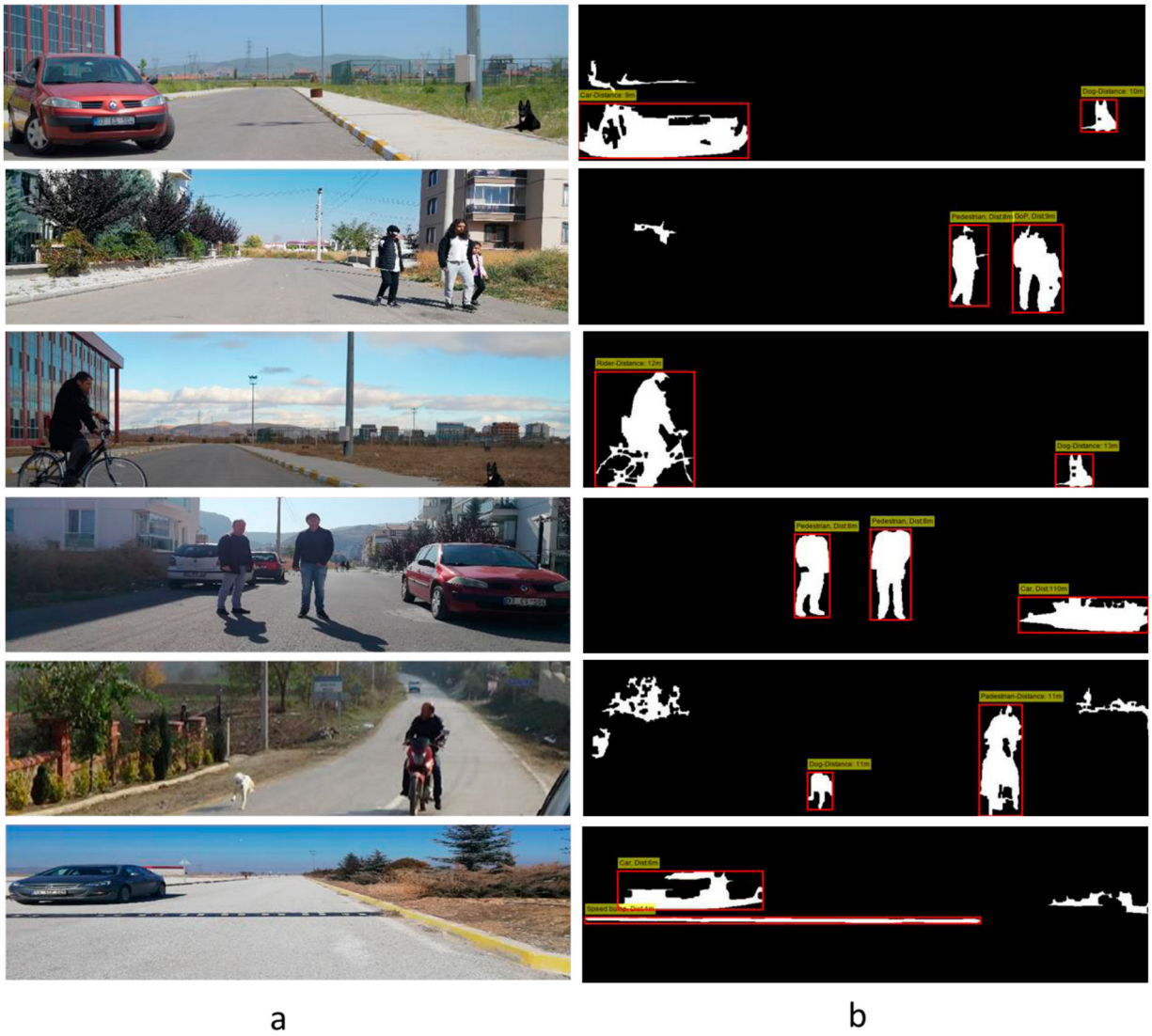

Figure 10. (a) Original images used in multiple object recognition, (b) object recognition results.

Table 8. YOLO vs ICANN performance comparison results.

	ICANN		YOLO v3	
Object Type	Accuracy (%)	Average Computational Time per Sample (ms)	Accuracy (%)	Average Computational Time per Sample (ms)
Pedestrian	87.4	16.2	81.3	17.8
GoP	83.1	17.1	79.0	21.2
Speed Bump	94.8	13.6	80.2	21.2
Car	90.4	15.9	88.7	16.1
Rider	83.6	17.1	81.1	19.8
Pothole	71.8	14.2	44.8	22.8
Dog	86.8	17.2	86.1	19.9

There is no need for any pre-processing for object recognition in YOLO v3. In addition, the YOLO v3 algorithm can recognize more distant objects than ICANN. However, as seen in Table 8, when YOLO v3 was trained with the same number of samples, it was less successful than ICANN in terms of accuracy rate and processing time. The advantages and disadvantages of both methods within the scope of this study are listed below.

- In ICANN, objects are recognized using formal features over binarized images. Since YOLO v3 uses RGB images and deep neural networks for training, a lot of images are required, and training time is much longer than ICANN. The objects found in the images must be labelled one at a time for training. This is challenging because the objects' centre of gravity and the bounding box's information on width and length must be recorded into a text file.
- These comparisons did not include the newer YOLO V4 due to a few factors. One of the most important differences between the YOLO V4 and the YOLO v3 is that the YOLO V4 can produce accurate results at higher fps. Because only 30 fps are employed in this study, this advantage is ineffective. Another benefit is an increase in the mean average precision (mAP) value. However, for YOLO v3 and v4 to achieve high mAP values, extensive training with a large number of samples is required. For example, in the training processes with YOLO v4, 2594 sample images were used by

Figure 11. Recognized objects using (a) ICANN, (b) YOLO v3.

Albahli et al. [55] and 70000 by Li et al. [56]. Therefore, when the number of samples used for training is low, as in this study, YOLO v4 has no advantage.

- The longest training period in ICANN is 8.1 s for three hidden layers composed of 10 neurones. YOLO v3 runs out approximately 39 billion operations for one image. It then needs high-capacity computer systems [57,58].
- The YOLO v3 algorithm can recognize more distant objects than ICANN in images with complex backgrounds, but this often turns into a disadvantage. Errors such as recognizing a car as a truck, shoes as a skateboard, or detecting three pedestrians instead of two can occur.
- When using the whole weight file, CNN-based methods also try to recognize unnecessary objects in the background. However, ICANN focuses only on the specified objects.
- Since the background images of the route that do not contain any foreign objects (reference images) are stored in the system memory in the proposed method, modifications in the backgrounds over time can also be detected. There is no such possibility in CNN-based methods.

When the literature was reviewed, no other study was found that used all the objects we classified. Instead, there are individual pedestrian, bicycle, or car recognition studies. For example, in the study where only pedestrian recognition was performed by Lan et al. [3], successful results were obtained with 88.71% with YOLO and 80.11% with ConvNet. In another pedestrian recognition study conducted by Kuang et al. [5], a precision rate of 76.8% with YOLO and 65.14% with R-CNN was achieved. Benjdira et al. [59] achieved 98.8% with YOLO and 65.4% with Faster R-CNN in their car recognition study. In a study performed by Sarıbaş et al. [60] in which unmanned aerial vehicles were used, an 83.5% success rate and 0.136 s processing time were achieved with YOLO. When the results are compared, it is realized that our successful recognition rates are greater than most research. However, it should be underlined that other studies were performed using neural networks trained for only a single object.

6. Conclusion

In this study, a system has been developed for a UGV to detect and recognize the obstacles encountered during

its real-time movement on predetermined routes. The system uses images consisting of blank backgrounds previously taken from the geographical location and stored in the memory for detection. Due to the computational cost advantage of this situation, the time limits required for object detection and recognition are not exceeded. Interconnected ANN sequences are used for object recognition. The object/noise classification training performed in the first step of ICANN achieved 98.5% accuracy. In fact, the success of this classification significantly affects the success and speed of the whole system. In the MOD step, the morphological filtering steps that increase the computational cost have been neglected due to the success of the object/noise classification. Therefore, the real-time processing speed has increased.

Accurate recognition rates are average 91.2% for single objects. However, this rate drops to 86.5% in Multi-Object Recognition. This decrease is due to the lack of desired quality of the snapshot taken from the reference coordinates due to environmental conditions. Due to camera shake, shadow effect, and light irregularities, sometimes the desired quality of the image could not be obtained. In order to reduce the negative effect of the illumination on the image quality, photo resistive sensors were used, and different threshold values were determined at any time of the day. In addition, the results obtained in the tests carried out at 4 km/h speed are lower than the other speed. One of the reasons is thought to be vibration. Another reason is that some reference coordinates are missed due to speed. Although this situation decreases the success rate, the objects can be detected successfully after the missed reference coordinate.

As a result, when the experimental results of the developed system were examined, the average success rates for single objects were 91.2% at 3 km/h and 88.3% at 4 km/h. For multiple objects, it was 86.5% and 82.6%, respectively. The developed system provides cost savings since these operations are performed using only the camera. Successful classification rates can be increased by increasing the number of samples in ICANN training processes. In order to increase the speed of UGV, it is necessary to use computer systems with higher CPU and GPU capacity. Roads and roadside traffic signs will be added to the classes that will be recognized in future studies. In the next stage of this study, it is planned to perform simultaneous localization and mapping (SLAM) study in non-GPS environments using pre-recorded images.

Acknowledgments

This research was supported by Afyon Kocatepe University, Scientific Research Projects Council with Project Number 14.FEN.BİL.37.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

This research was supported by Afyon Kocatepe Üniversitesi, Scientific Research Projects Council with Project Number 14.FEN.BİL.37.

ORCID

Barış Gökçe http://orcid.org/0000-0001-6141-7625
Güray Sonugür http://orcid.org/0000-0003-1521-7010

References

[1] Rashid M, Khan MA, Alhaisoni M, et al. A sustainable deep learning framework for object recognition using multi-layers deep features fusion and selection. Sustainability. 2020;12:5037), https://doi.org/10.3390/SU12125037.
[2] Masood H, Zafar A, Ali MU, et al. Recognition and tracking of objects in a clustered remote scene environment. computers. Materials & Continua. 2020;70:1699–1719. https://doi.org/a Tech Science Press DOI: 10.32604/cmc.2022.019572.
[3] Lan W, Dang J, Wang Y, et al. Pedestrian detection based on yolo network model). Proceedings of 2018 IEEE International Conference on Mechatronics and Automation, ICMA 2018 1547–1551; 2018. https://doi.org/10.1109/ICMA.2018.8484698
[4] Rashid M, Khan MA, Sharif M, et al. Object detection and classification: a joint selection and fusion strategy of deep convolutional neural network and SIFT point features. Multimed Tools Appl. 2018;78(12):15751–15777. https://doi.org/10.1007/S11042-018-7031-0.
[5] Kuang P, Ma T, Li F, et al. Real-time pedestrian detection using convolutional neural networks. Int J Pattern Recognit Artif Intell. 2018;32:1–16. https://doi.org/doi:10.1142/s0218001418560141.
[6] Demim F, Nemra A, Louadj K, et al. Cooperative SLAM for multiple UGVs navigation using SVSF filter. Automatika. 2017;58:119–129. https://doi.org/10.1080/00051144.2017.1372123.
[7] Morales N, Toledo JT, Acosta L, et al. Real-time adaptive obstacle detection based on an image database. Comput Vis Image Underst. 2011;115:1273–1287. https://doi.org/10.1016/j.cviu.2011.05.004.
[8] He R, Wei R, Zhang Q. UAV autonomous collision avoidance approach. Automatika. 2017;58:195–204. https://doi.org/10.1080/00051144.2017.1388646.
[9] Raparelli E, Bajocco S. A bibliometric analysis on the use of unmanned aerial vehicles in agricultural and forestry studies. Int J Remote Sens. 2019;40:9070–9083. https://doi.org/10.1080/01431161.2019.1569793.
[10] Kakani V, Kim H, Kumbham M, et al. Feasible self-calibration of larger field-of-view (FOV) camera sensors for the advanced driver-assistance system (ADAS). Sensors (Switzerland. 2019;19:1–29. https://doi.org/10.3390/s19153369.
[11] Van Hien N, Van He N, Diem PG. A model-driven implementation to realize controllers for Autonomous Underwater Vehicles. Appl Ocean Res. 2018;78:307–319. https://doi.org/10.1016/j.apor.2018.06.020.
[12] Masoud O, Papanikolopoulos NP. A novel method for tracking and counting pedestrians in real-time

using a single camera. IEEE Trans Veh Technol. 2001;50:1267–1278. https://doi.org/10.1109/25.950328.
[13] Del BA, Dini F, Grifoni A, et al. Uncalibrated framework for on-line camera cooperation to acquire human head imagery in wide areas). Proceedings - IEEE 5th International Conference on Advanced Video and Signal Based Surveillance, AVSS 2008 252–258; 2008. https://doi.org/10.1109/AVSS.2008.69
[14] Kanat ÖÖ, Karatay E, Köse O, et al. Combined active flow and flight control systems design for morphing unmanned aerial vehicles. Proceedings of the Institution of Mechanical Engineers. Part G: Journal of Aerospace Engineering. 2019;233:5393–5402. https://doi.org/10.1177/0954410019846045.
[15] Thangaraj M, Monikavasagom S. A competent frame work for efficient object detection, tracking and classification. Wirel Pers Commun. 2019;107:939–957. https://doi.org/10.1007/s11277-019-06310-4.
[16] Tadokoro S, Murphy R, Stover S, et al. Application of active scope camera to forensic investigation of construction accident. Proceedings of IEEE Workshop on Advanced Robotics and its Social Impacts, ARSO. 2009;1:47–50. https://doi.org/10.1109/ARSO.2009.5587076.
[17] Ghobadi SE, Hartmann K, Weihs W, et al. Detection and classification of moving objects-stereo or time-of-flight images. 2006 International Conference on Computational Intelligence and Security, ICCIAS. 2007; 2006 (1):11–16. https://doi.org/10.1109/ICCIAS.2006.294082.
[18] Gonzalez JP, Ozguner U. Lane detection using histogram-based segmentation and decision trees. IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC. 2000;1:346–351. https://doi.org/10.1109/ITSC.2000.881084.
[19] Hussain N, Khan MA, Kadry S, et al. Intelligent deep learning and improved whale optimization algorithm based framework for object recognition. Human-centric Computing and Information Sciences. 2021;11: 1–17. https://doi.org/10.22967/HCIS.2021.11.034.
[20] Mejia V, Kang EY. Automatic moving object detection using motion and color features and bi-modal Gaussian approximation. Conference Proceedings - IEEE International Conference on Systems, Man and Cybernetics. 2011;1:2922–2927. https://doi.org/10.1109/ICSMC.2011.6084109.
[21] Hayman E, Eklundh JO. Statistical background subtraction for a mobile observer. Proceedings of the IEEE International Conference on Computer Vision. 2003;1:67–74. https://doi.org/10.1109/iccv.2003.1238315.
[22] Panda DK, Meher S. Adaptive spatio-temporal background subtraction using improved Wronskian change detection scheme in Gaussian mixture model framework. IET Image Proc. 2018;12:1832–1843. https://doi.org/10.1049/iet-ipr.2017.0595.
[23] Norouzi Sefidmazgi A, Nahvi M. Improved background modeling of video sequences using spatio-temporal extension of fuzzy local binary pattern. Multimed Tools Appl. 2019;78:17287–17316. https://doi.org/10.1007/s11042-018-6972-7.
[24] Yu Y, Kurnianggoro L, Jo KH. Moving object detection for a moving camera based on global motion compensation and adaptive background model. International Journal of Control, Automation and Systems. 2019;17:1866–1874. https://doi.org/10.1007/s12555-018-0234-3.
[25] Kong H, Kim J, Ye G, et al. Moving object detection under free-moving camera. Image Process. 2010: 4669–4672.
[26] Liu H, Hang TH, Herman M, et al. Accuracy vs. Efficiency trade-offs in optical flow algorithms. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). 1996;1065:174–183. https://doi.org/10.1007/3-540-61123-1_137.
[27] Ren Y, Chua CS, Ho YK. Statistical background modeling for non-stationary camera. Pattern Recognit Lett. 2003;24:183–196. https://doi.org/10.1016/S0167-8655(02)00210-6.
[28] Minaeian S, Liu J, Son YJ. Effective and efficient detection of moving targets from a UAV's camera. IEEE Trans Intell Transp Syst. 2018;19:497–506. https://doi.org/10.1109/TITS.2017.2782790.
[29] Afaq Ali Shah S, Bennamoun M, Boussaid F. Iterative deep learning for image set based face and object recognition. Neurocomputing. 2016;174:866–874. https://doi.org/10.1016/j.neucom.2015.10.004.
[30] He T, Mao H, Yi Z. Moving object recognition using multi-view three-dimensional convolutional neural networks. Neural Computing and Applications. 2017;28:3827–3835. https://doi.org/10.1007/s00521-016-2277-9.
[31] Ullah A, Wang J, Anwar MS, et al. Fusion of machine learning and privacy preserving for secure facial expression recognition. Security and Communication Networks. 2021;2021:1–12. https://doi.org/10.1155/2021/6673992.
[32] Gu Z, Nazir S, Hong C, et al. Convolution neural network-based higher accurate intrusion identification system for the network security and communication. Security and Communication Networks. 2020;2020, https://doi.org/10.1155/2020/8830903.
[33] Mukhtar N, Khan MA, Chiragh N, et al. Recognition and Effective Handling of Negations in Enhancing the Accuracy of Urdu Sentiment Analyzer. Mehran Univ Res J Eng Technol. 2020;39:759–771. https://doi.org/10.22581/muet1982.2004.08.
[34] Girshick R, Donahue J. TD-P of the, 2014 U Rich feature hierarchies for accurate object detection and semantic segmentation). In: IEEE Conference on Computer Vision and Pattern Recognition. Colombus (OH) / USA; 2014. pp 580–5587
[35] Girshick R. Fast R-CNN. Proceedings of the IEEE International Conference on Computer Vision 2015 Inter. 2015;1:1440–1448. https://doi.org/10.1109/ICCV.2015.169.
[36] Ren S, He K, Girshick R, et al. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. IEEE Trans Pattern Anal Mach Intell. 2017;39:1137–1149. https://doi.org/10.1109/TPAMI.2016.2577031.
[37] Redmon J, Divvala S, Girshick R, et al. You Only Look Once: Unified, Real-Time Object Detection). In: In Proceedings of the IEEE conference on computer vision and pattern recognition; 2016. pp 779–788
[38] Liu W, Anguelov D, Erhan D, et al. SSD: Single shot multibox detector. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). 2016;9905 LNCS:21–37. https://doi.org/10.1007/978-3-319-46448-0_2.
[39] Dewi C, Chen RC, Liu YT, et al. Yolo V4 for Advanced Traffic Sign Recognition with Synthetic

Training Data Generated by Various GAN. IEEE Access. 2021;9:97228–97242. https://doi.org/10.1109/ACCESS.2021.3094201.

[40] Tai S-K, Dewi C, Chen R-C, et al. Deep Learning for Traffic Sign Recognition Based on Spatial Pyramid Pooling with Scale Analysis. Applied Sciences. 2020;10:6997. https://doi.org/10.3390/APP10196997.

[41] Ahmad T, Ma Y, Yahya M, et al. Object Detection through Modified YOLO Neural Network. Sci Program. 2020;2020:1–10. https://doi.org/10.1155/2020/8403262.

[42] Kong H, Audibert JY, Ponce J. Detecting abandoned objects with a moving camera. IEEE Trans Image Process. 2010;19:2201–2210. https://doi.org/10.1109/TIP.2010.2045714.

[43] Yoon H, Hoskere V, Park J-W, et al. Cross-Correlation-Based Structural System Identification Using Unmanned Aerial Vehicles. Sensors. 2017;17:2075), https://doi.org/10.3390/s17092075.

[44] Karpenko A, Jacobs D, Baek J, et al. Digital Video Stabilization and Rolling Shutter Correction using Gyroscopes. Stanford Tech Report CTSR. 2011;1(3):1–7.

[45] Hussain N, Khan MA, Sharif M, et al. A deep neural network and classical features based scheme for objects recognition: an application for machine inspection. Multimed Tools Appl. 2020;2020:1–23. https://doi.org/10.1007/S11042-020-08852-3.

[46] Levenberg K. A method for the solution of certain non-linear problems in least squares. Q Appl Math. 1944;2:164–168. https://doi.org/10.1090/QAM/10666.

[47] Marquardt D. An Algorithm for Least-Squares Estimation of Nonlinear Parameters. SIAM J Appl Math. 1963;11:431–441.

[48] Moller MF. A scaled conjugate gradient algorithm for fast supervised learning. Neural Netw. 1993;6:525–533.

[49] Cheraghi SA. (2012). Moving Object Detection Using Image Registration for a Moving Camera Platform. 23–25.

[50] Bay H, Tuytelaars T, Van Gool L. SURF: Speeded Up Robust Features.

[51] Harris C, Stephens M. A Combined Corner and Edge Detector. Procedings of the Alvey Vision Conference. 1988;1988(6):1–23. https://doi.org/10.5244/C.2.23.

[52] Rosten E, Drummond T. Machine learning for high-speed corner detection. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). 2006;3951 LNCS:430–443. https://doi.org/10.1007/11744023_34.

[53] Gonzalez RC, Woods RE, Eddins SL. Digital Image Processing using Matlab, Second edi. USA: Gatesmark; 2011.

[54] Jampana P, Shah S. An image differencing method for interface level detection in separation cells. Mach Vis Appl. 2012;23:283–298. https://doi.org/10.1007/s00138-010-0306-8.

[55] Albahli S, Nida N, Irtaza A, et al. Melanoma Lesion Detection and Segmentation Using YOLOv4-DarkNet and Active Contour. IEEE Access. 2020;8:198403–198414. https://doi.org/10.1109/ACCESS.2020.3035345.

[56] Li Y, Wang H, Dang LM, et al. A deep learning-based hybrid framework for object detection and recognition in autonomous driving. IEEE Access. 2020;8:194228–194239. https://doi.org/10.1109/ACCESS.2020.3033289.

[57] Goel A, Tung C, Lu Y-H, et al. (2020). A Survey of Methods for Low-Power Deep Learning and Computer Vision.

[58] Nguyen DT, Nguyen TN, Kim H, et al. A high-throughput and power-efficient fpga implementation of yolo cnn for object detection. IEEE Transactions on Very Large Scale Integration (VLSI) Systems. 2019;27:1861–1873. https://doi.org/10.1109/TVLSI.2019.2905242.

[59] Benjdira B, Khursheed T, Koubaa A, et al. Car Detection using Unmanned Aerial Vehicles: Comparison between Faster R-CNN and YOLOv3). 2019 1st International Conference on Unmanned Vehicle Systems-Oman, UVS 2019; 2019. https://doi.org/10.1109/UVS.2019.8658300

[60] Saribas H, Cevikalp H, Kahvecioglu S. Car detection in images taken from unmanned aerial vehicles. 26th IEEE Signal Processing and Communications Applications Conference, SIU. 2018;2018 :1–4. https://doi.org/10.1109/SIU.2018.8404201.