Machine Learning with Applications 6 (2021) 100104

Contents lists available at ScienceDirect

Machine Learning with Applications

journal homepage: www.elsevier.com/locate/mlwa

A deep learning based hazardous materials (HAZMAT) sign detection robot with restricted computational resources☆

Amir Sharifi [a,1], Ahmadreza Zibaei [a,1], Mahdi Rezaei [b,1,*]

[a] Advanced Mobile Robotics Lab, Qazvin Islamic Azad University, Qazvin 13419, Iran
[b] Institute for Transport Studies, University of Leeds, 34-40 University Road, Leeds, LS2 9JT, United Kingdom

ARTICLE INFO

Keywords:
Hazardous materials
Object recognition
HAZMAT sign detection
Segmentation
CNN
Rescue robotics

ABSTRACT

One of the most challenging and non-trivial tasks in robot-based rescue operations is the Hazardous Materials (HAZMAT) sign detection in dangerous operation fields, in order to prevent further unexpected disasters. Each HAZMAT sign has a specific meaning that the rescue robot should detect and interpret to take a safe action, accordingly. Accurate HAZMAT detection and real-time processing are the two most important factors in such robotics applications. Furthermore, the rescue robot should cope with some secondary challenges such as image distortion and restricted CPU and computational resources, embedded in the robot. In this research, we propose a CNN-Based pipeline called DeepHAZMAT for HAZMAT sign detection and segmentation in four steps: (1) Input data volume optimisation before feeding into the CNN network, (2) Application of a YOLO-based structure to collect the required visual information from the hazardous areas, (3) HAZMAT sign segmentation and separation from the background using adaptive GrabCut technique, and (4) Post-processing optimisation using morphological operators and convex hull algorithms. In spite of the utilisation of a very limited CPU and memory resources, the experimental results show the proposed method has successfully maintained a better performance in terms of detection-speed and detection-accuracy, compared to classical and modern state-of-the-art methods.

1. Introduction

According to the U.S. Department of Transport (DOT) guidelines (Registei, 2012), all vehicles that transport dangerous goods or HAZMAT materials/parcels must use large and clear HAZMAT signs in front and back of the vehicles, to indicate the type of hazard for other road users. These signs identify information described by the sign shape, colour, symbols, and numbers. Fig. 1 shows samples of HAZMAT signs (Anderson, 2018). Because of the importance of HAZMAT signs, all of them are globally accepted; therefore, each sign is language-independent. Every sign clearly describes the hazard type; so, the rescue team will be prepared to take action for the required tools and equipment to cope with the challenge.

The 2011 Fukushima Nuclear Power Plant disaster in Japan (Hirose, 2012), draw the attentions of lots of researchers to the importance of developing autonomous rescue robots for complex and dangerous tasks (Ohno et al., 2011). The recent development of Computer Vision and deep learning techniques has significantly helped to speed up the research (Rezaei & Klette, 2017). In 2016, the RoboCup international committee officially announced the HAZMAT sign detection as one of the main streams of competitions among rescue robot teams (RoboCup Comittee, 2020). HAZMAT sing-related research currently is in its infancy with a very high research potential to discover the existing challenges and opportunities.

Computer vision is employed for object detection and recognition in various environments. In some robots, vision is used to detect and localise different objects Hughes et al. (2019), and HAZMAT sign detection in rescue scenarios is a challenging computer-vision based task due to environmental situations such as various lighting conditions, image perspective distortion, camera angles, image blurring, and frequent contrast changes.

A rescue robot has a limited power supply, memory and computational resources; and on the other hand, it needs to perform a real-time, accurate, and reliable sign detection to save lives.

Speed and accuracy are always contradictory to each other. i.e. developing a more accurate system normally decreases the speed and vice

☆ Real-time and accurate HAZMAT recognition is an important challenge in rescue robotics operations.
The code (and data) in this article has been certified as Reproducible by Code Ocean: (https://codeocean.com/). More information on the Reproducibility Badge Initiative is available at https://www.elsevier.com/physical-sciences-and-engineering/computer-science/journals.

* Corresponding author.
E-mail addresses: amir.sharifi@qiau.ac.ir (A. Sharifi), a.zibaei@qiau.ac.ir (A. Zibaei), m.rezaei@leeds.ac.uk (M. Rezaei).

[1] All authors have contributed equally.

https://doi.org/10.1016/j.mlwa.2021.100104
Received 15 November 2020; Received in revised form 29 June 2021; Accepted 30 June 2021
Available online 6 July 2021

Fig. 1. Samples of standard and globally recognised Hazardous materials (HAZMAT) signs.

versa. So we need to consider a trade-off in between, which makes the task even more difficult.

This paper proposes a real-time method that provides a fast HAZMAT sign detection, while maintaining a high-level accuracy in various lighting conditions and different backgrounds. Furthermore, the system can be implemented on a small low weight mobile robots with limited memory, hardware and computational resources.

In the following sections, we discuss more in-depth and provide further details. The rest of the paper is organised as follows: Section 2 reviews the literature and related works. A comparison between two mainstream models is conducted in Section 3. In Section 4 we introduce the characteristics of our rescue robot which is developed to perform in real-world life saving scenarios. Then we provided the details of the proposed methodology in Section 5 The outcome of the experimental results will be discussed in Section 6 and finally, Section 7 summarises the paper with concluding remarks.

2. Related work

The literature review and common approaches in HAZMAT sign detection can be categorised in five main categories:

2.1. Colour-based methods

Colour-based methods mostly try to find the candidates' region of interest based on the colours of a HAZMAT sign and then pass them to a conventional key feature extractor such as SIFT (Lowe, 2004a) or SURF Based on research by Gossow et al. (2008), the accuracy of colour-based techniques may significantly decrease at the distances of 1.5 m and 2.0 m to as low as 52% and 20%, respectively. Sensitivity to lighting conditions and illumination changes are two other weaknesses of the colour-based methods.

2.2. Shape-based methods

These methods try to find the candidate regions of interest using the appearance and shape characterise of each HAZMAT shape. First, they generate the edge map of the region of interest (ROI) and then aim to find the HAZMAT attributes using shape-line attributes methods such as line or circle Hough transform (Loy & Barnes, 2004; Pao et al., 1992). These approaches may also find the candidate shape based on the SVM classifiers (Maldonado-Bascon et al., 2007) after an initial content recognition, or based on Gaussian-kernel SVMs. Shape-based methods are not occlusion-invariant, and also very sensitive to perspective distortions.

2.3. Saliency-based methods

Saliency-based methods aim to detect the silence objects, and then highlight and export them as the candidate regions. After that the accurate region of the sign will be detected using SIFT, SURF, ORB, FREAK, or other detection methods (Itti et al., 1998; Parra et al., 2013a; Parra et al., 2013b; Won et al., 2018).

Most of the above-discussed techniques have the following weaknesses in real-world scenarios:

1. In cases where the HAZMAT labels were arranged too close to each other, the detection process may completely fail.
2. Complex background may also cause HAZMAT signs detection failure.
3. The system only detects the HAZMAT signs from a very limited viewing angle.

2.4. Keypoint matching based methods

SURF is a common keypoint detector that is invariant to image scaling and rotation which has been built based on the widely used SIFT detector; The SURF is well-known for its faster processing comparing to SIFT. Using integral images, the SURF algorithm applies average filters instead of the Gaussian filters in SIFT detector. Edlinger et al. (2019) The speed-up process plays an important role in the cases where the detector method must be fast enough to be considered real-time. The OpenCV's contribution modules provide some of the common keypoint detectors such as SIFT (Lowe, 2004a), SURF (Gossow et al., 2008), and ORB which we will evaluate them in this research. In order to detect HAZMATs in real-time, every candidate was passed to the keypoint detector. Then a keypoint matcher between the keypoint database and the detected keypoints were used to recognise the object. The real-time performance of the algorithms such as SURF is expected to be satisfying; however, we need to ensure for the robustness of detection. We evaluate this in the section Experimental results.

2.5. Deep learning based methods

Due to the discussed weaknesses of the conventional object detection techniques (Sabzevari et al., 2008) as well as manual image engineering requirements, the research approaches has been redirected to modern deep learning based techniques. In deep neural network based methods the models try to extract some common features of the HAZMAT sign during the training phase.

In Edlinger et al. (2019), the authors aim to train a deep neural network model; however, the proposed model fails to detect HAZMAT signs in complex backgrounds.

Nils et al. Tijtgat et al. (2017), train a deep neural network model based on the YOLOv2 algorithm for HAZMAT sign detection. Although their model performs fast on a GPU platform, their model is not real-time on CPUs and the system has an error of up to 1.5″ in localising the HAZMAT signs.

Another recent approach in this field is proposed in Cai et al. (2020). In this method the system receives the visual and depth data of the environment by utilisation of an RGB-D camera. The proposed method computes the Homography to transfer between 3D and 2D perspective images. Then they rectify the image to cope with the distortion effects and the resulted images are feed to a CNN detector. Using the Homography matrix the system can also calculate the angle of the objects with respect to the camera coordinate system. The advantage of this approach is the ability of detecting the HAZMAT signs at various angles; yet it requires a very high computational cost. Therefore, this system is also not feasible for real-time performances. Furthermore, it needs additional sensors than common HAZMAT detection systems.

Most of the mentioned methods require heavy processing or are not accurate enough. For example, key point-based methods are strongly

influenced by environmental factors. If the region of interest (HAZMAT sign) appears in a complex background or the HAZMAT sign is polluted with oil, dirt, or dust, many extra keypoints will be detected which make it difficult to detect the main keypoints to find and detect the actual object in the image. This may increase the detection error rate; however, deep learning based methods, on the other hand, can easily overcome this problem by learning complex examples and scenarios during the network training phase.

As a common weakness of the deep learning based methods, they mainly focus to increase the accuracy and do not consider the limited resources of a rescue robot which normally performs on a compact embedded CPU platform.

In contrast to the above methods, and in order to cope with the aforementioned bottlenecks, this article adapts and examines a specific deep neural network model that can be used in low performance systems with a few simple yet efficient hierarchical techniques.

3. Model comparison: YOLO vs. conventional methods

In this section, we briefly justify our hypothesis in the effectiveness of using a YOLO-based framework for HAZMAT sign detection and compare it with conventional methods.

Unlike the common deep learning based models, YOLO (You Only Look Once) family (Redmon et al., 2016) object detectors do not use common sliding windows. This causes the YOLO algorithm to run just once for each image and becomes one of the fastest object detection algorithms.

In Section Experimental Results, we examine various types of deep convolutional networks including YOLO family and EfficientDet, as well as the conventional feature point-based SIFT method. We hypothesise and investigate if the YOLOv3 tiny (as one of the cheapest computational models), can be an option as a fast and accurate solution for our application. YOLOv3 tiny algorithm takes the following steps to detect a HAZMAT sign:

- Dividing the input image into cells (grids)
- Performing bounding box prediction for each grid cell
- Eliminating low probability predictions
- Applying non-maximum suppression to generate final predictions

The method divides every input image into an $S \times S$ grid of cells and each grid predicts B anchor boxes, C classes, and probabilities of the objects that their centres fall inside the grid cells. We adapt the YOLOv3-tiny to detect a HAZMAT sign in two different scales, in order to accommodate different objects sizes. In our model we feed input images 576×576 pixels, and the YOLOv3-tiny will makes the detection on the scale of 13×13 and 26×26. Fig. 2, the right column shows some initial and raw outcomes of a YOLO based Hazmat detection.

On the other hand, Fig. 2, the left column, shows the detection results of SIFT using keypoints features by adapting the Lowe's paper (Lowe, 2004b). SIFT uses feature descriptors which are invariant against various transformations.

Considering the reviewed related works in the previous section and existing research gaps and challenges, we will offer a new model with three main contributions as follows:

(1) We publicly release a standard HAZMAT dataset with PASCAL-VOC format as a new comprehensive dataset to be used by other researchers in the field.

(2) We introduce a CNN-based neural network model for HAZMAT sign detection that successfully decreases the CPU usage by reducing the number of feed images into the network.

(3) Finally, we develop a custom Non-Maximum Suppression method that avoids a common issue of multiple false detections in other methods.

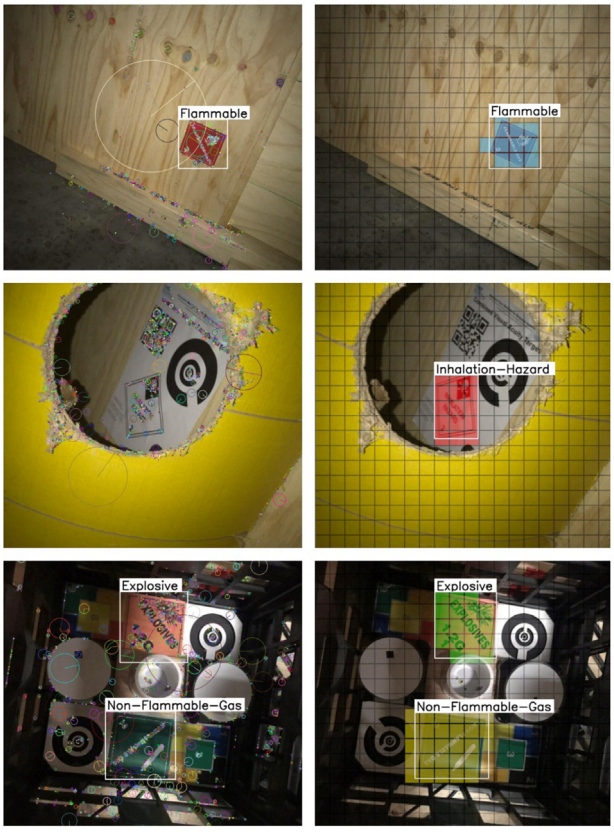


Fig. 2. Performance comparison: SIFT (left column) vs. YOLO (right column).

4. Rescue robot features

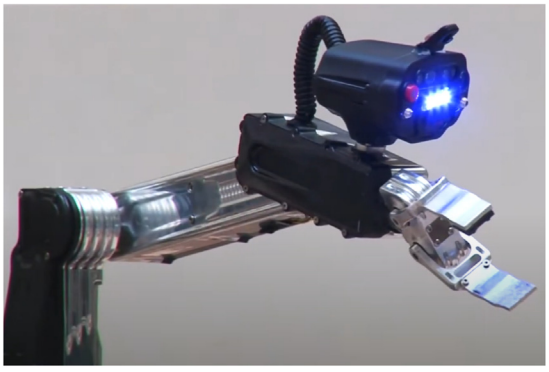
Before we dive into the details of our methodology, we would like to provide some further information about the role of rescue robots and our designed ARKA rescue robot in the Advanced Mobile Robotics Lab (AMRL). Rescue robots are designed to rescue people and/or provide environmental data to the rescue team in order to facilitate a rescue mission. The robots are mainly employed in extreme situations such as natural disasters, chemical/structural accidents, explosive detection, etc.

A rescue robot is a type of robot that can enter in dangerous disaster scenes and carry out rescue tasks on behalf of a human. Earthquake scenes, chemical sites, collapsed buildings and towers due to fire or explosion, are few examples that may take place on a daily basis all around the world. One of the most important factors in rescue operations is to find and save victims, in time.

4.1. ARKA rescue robot

ARKA is an unmanned ground vehicle (UGV) designed and developed by our team at Advanced Mobile Robotics Lab (Fig. 3). The ARKA is a transportable robot with superior mobility and advanced manipulation which is able to tackle dangerous situations. It weighs about 90 kg and can climb stairs up to a gradient of 45° and slopes of 55°. The robot is equipped with a dexterous manipulator, 360° rotating wrist, gripper, microphone, depth, and thermal cameras. The 13 kg manipulator can be extended up to a length of 140 cm. It can lift objects weighing more than 10 kg at full arm extension and roughly 35 kg at the close-in position. The ARKA is suitable for missions such as explosive detection, explosive ordnance disposal (EOD)/bomb disposal, persistent observation, gas leakage control, and vehicle inspections. The robot is very accurate and flexible in dealing with rescue and security

(a) Main body structure



(b) Dexterous manipulator, 360° rotating wrist, gripper, microphone, depth and thermal cameras

Fig. 3. ARKA rescue robot, developed in AMRL research lab.

problems, and it provides live video streams to map and localise in unknown environments (Najafi et al., 2019).

The robot is connected to an operator station via a 5 GHz WLAN. We used two computing platforms in our tests as follows: an Intel NUC mobile computer with an Intel Core i7-5557U processor for the robot, and another Intel NUC with a Core i5 processor for the remote operating station. No GPU is used on both sides.

The sensor box (Rezaei et al., 2010) is a crucial and essential hardware part of tactical, rescuer, and police mobile robots to increase their perception capabilities. In our ARKA robot, this pack is equipped with various types of sensors and devices as follows:

- **RGB-D Camera**: The Sensor Box is equipped with an Intel®RealSense™ depth-camera, which provides depth images and RGB images. One of the major usages of this sensor is to detect and avoid impassable grounds and obstacles, by using Point Clouds gained from the camera. Furthermore, to create a map of the current scene, the depth data is used for readiness tests in identification and dexterity operations.
  In order to detect any object in a rescue scenario, the main camera (Intel RealSense) is used to determine objects' locations in the 3D environment as well as obtaining the relative distance of the detected HAZMAT labels to the robot. Using the SLAM algorithm for self-localisation of the robot, this relative position can then be related to the created map.
- **CO2 sensor**: In order to find out whether the victim is breathing or not, an MQ-9 sensor is being used which can be customised for detecting other types of gases, if needed.
- **Thermal Camera**: One of the most important vital signs, for analysing whether the victim is still alive or not, is the temperature of the victim's body. Body position estimation and body temperature detection of the victim is accomplished by the combination of the previously discussed RGB-D sensor and equipping the autonomous robot with a Thermal Image sensor (miniAV160). This makes it capable of synchronous capturing of visual and thermal images.
- **Analog Cameras**: Two analogue cameras that are mounted on the Sensor Box, assists the operator to drive the robot and accomplish particular missions.
- **Input and Output Audio**: A microphone and a speaker is installed on the Sensor Box to provide full-duplex audio communication between the victim and the rescue team, at any time if needed.
- **Laser Scanner**: is used for some autonomous tasks, where we need to calculate the distance between the robot and obstacle(s). Moreover, for generating a 2D map of the environment we use a Hokuyo UTM30-LX LIDAR attached to a stabiliser. This way we guaranty that on sloping surfaces manoeuvres, the sensor stays parallel to the ground for a continuous mapping.

5. Methodology

In this section we discuss four major steps that we have taken to develop our methodology based on an appropriate DNN model: (A) Creating a training dataset, (B) implementing a customised Non-maximal suppression function (C) Data Feeding Optimisation, and (D) Data Logging.

5.1. Dataset

One of the major challenges in deep learning based methodologies is the requirement of large training datasets in order to achieve excellent results (Rezaei & Shahidi, 2020).

Convolutional Neural Networks or CNNs are supervised learning approaches, i.e. the labelled images that constitute as ground truth data must be initially provided to train the neural network. Preparing a good dataset is as important as a good neural network structure. Having no properly engineered dataset, it is very unlikely to get the maximum performance out of a network.

We have developed the HAZMAT-13 dataset, a comprehensive dataset of HAZMAT signs, including 1685 images from various viewing angles, distortions, and different illumination conditions. The dataset has been divided into 13 different classes as per Table 1.

The dataset is annotated with PASCAL-VOC format as it is easy to convert into other annotation formats such as YOLO or COCO. Furthermore, the dataset can be easily labelled using the labelImg (Tzutalin, 2020) labelling tool.

Fig. 4 shows some HAZMAT sign samples from the HAZMAT-13 dataset. Applying data augmentation we increased the number of HAZMAT signs to improve the performance of our algorithm. Besides, the dataset needs to be balanced and the number of images for each class should be almost the same to have a Homogeneous dataset. Also, the size of our dataset should neither be very small that lead to model under-fitting and detection accuracy loss, and nor too large to increase the complexity of the feature extraction and overfitting challenges. To aim this, and using the augmentation technique, we expanded the dataset to 4065 images per class and in overall 52845 images. We split the dataset into 80% train set and 20% test set.

For choosing our CNN model we had to consider a couple of more factors:

1. To be fast enough and implementable on mobile robots with restricted CPU resources,
2. To be accurate enough to get one of the best detection performances among state-of-the-arts.

YOLO is one of the most common object detection networks, which can be appropriately reconfigured by changing the size and the number of layers to satisfy a trade-off between speed and accuracy for our model. Since we are limited to use a low power CPU, we deploy a lighter version of YOLO, named YOLOv3-tiny (Redmon & Farhadi, 2018). Fig. 5 illustrates the architecture of our improved YOLOv3-tiny model. We provide further details in the next three subsections.

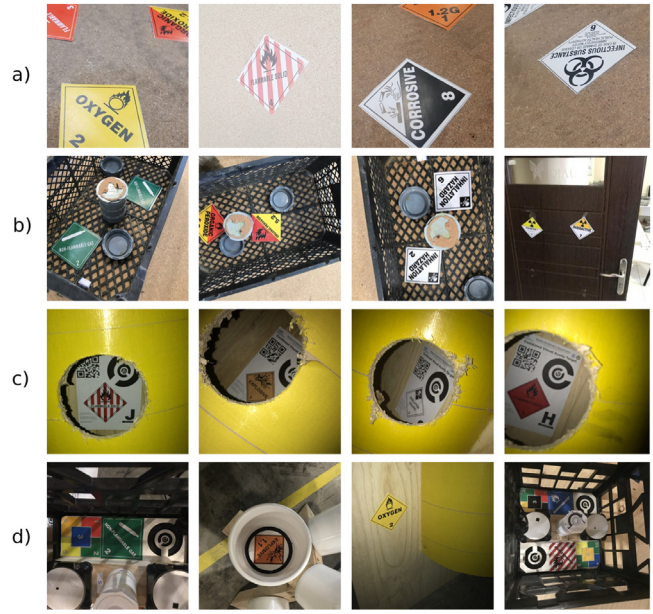


Fig. 4. Row a, b: Samples in different angles, lighting conditions, and backgrounds. Row c, d: Samples in different situations at RoboCup competitions in past years.

Table 1
13 categories of HAZMAT signs provided in the DeepHAZMAT dataset.

1- Poison	
2- Oxygen	
3- Flammable Gas	
4- Flammable Solid	
5- Corrosive	
6- Dangerous	
7- Non-flammable Gas	
8- Organic Peroxide	
9- Explosive	
10- Radioactive	
11- Inhalation Hazard	
12- Spontaneously Combustible	
13- Infectious Substance	

5.2. Image feeding optimisation

Many service robots use landmarks and GPS location data to improve and speed up the search processes for the target objects in the scene (Kim & Suh, 2019). In rescue robot operations, the HAZMAT signs can be anywhere, on the ground, walls, or windows, which makes the search process more time consuming and challenging.

In a live video sequence, the majority of the image frames may not include any HAZMAT signs, so it would be wise that we only look for the image frames which more likely contain a HAZMAT sign, rather than searching the entire frame sequences. This will significantly reduce the CPU usage. As a brief overview of our design, we perform a quick search for HAZMAT signs only within some of the input frames (initially, let us say on 50% of the input frames depending on the camera frame rate) until we notice that some regions of a specific frame can be candidate regions of interest. Then we focus on more

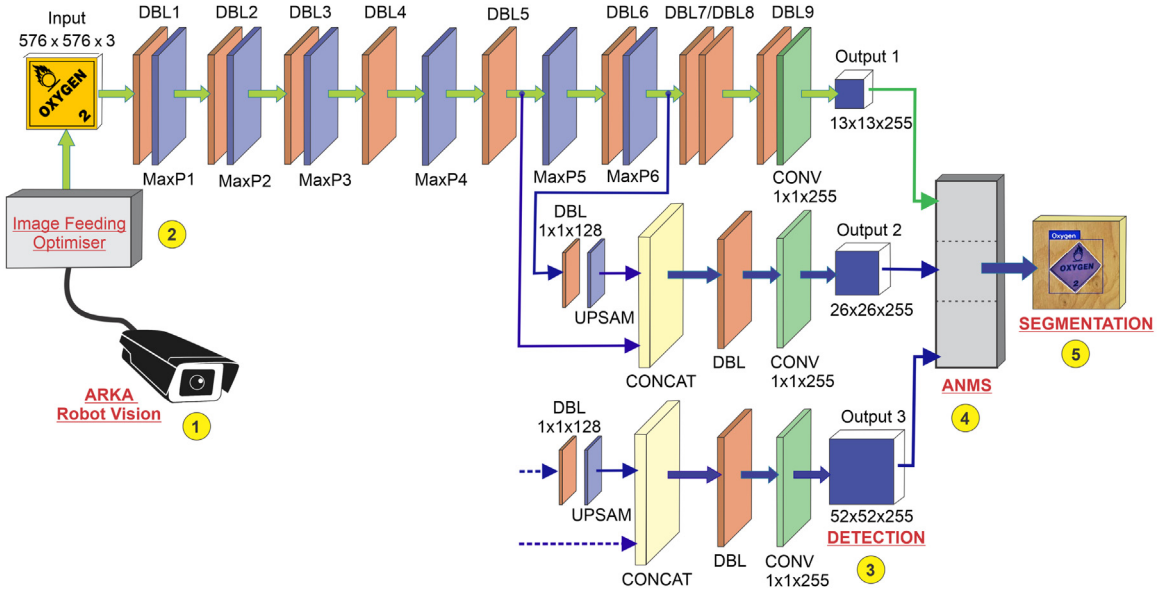


Fig. 5. Structure of the five main modules of the developed HAZMAT recognition system including (1) ARKA Robot Vision module, (2) Image Feeding Optimiser, (3) YOLOv3-tiny based HAZMAT detector, (4) Adaptive Non-maximal Suppression module, and (5) Segmentation module.

Algorithm 1: Increase/Decrease skip frames value

```
k := 5;    q := 2^k
p := q;    n := 0
while hasNewFrame do
    n = n + 1
    if n > p then
        n = 0
        frame := getFrame()
        objects := detectHazmats(frame)
        if len(objects) > 0 then
            if p > 1 then
                p = p / 2
            end
        else
            if p < q then
                p = p * 2
            end
        end
    end
end
```

consecutive frames at those ROIs to recognise and verify the HAZMAT signs and their types.

Algorithm 1 provides further details about our approach. The action is similar to the way a human searches for recognising particular objects in an unknown environment. A human first scans the entire environment quickly, and then if during the scan process, his/her attention drawn to a particular region, he/she will focus on the ROI with more concentration and accurate search to recognise the object.

In Algorithm 1, we have two main parameters p and q. If we assume S_c as the camera speed in frames per second (fps), then we set $q = 2^k$ where $k \in \mathbb{Z}$ and q is the smallest squared integer number which is greater than S_c, so that the quotient of the division operation $q \div S_c$ becomes greater than 1. For example, if the camera speed is 30 fps then k would be equal to 5, because $q = 2^5 = 32$ and the quotient of $32 \div 30 = 1$. In other words, we initially only analyse one frame in every second to find HAZMAT signs (1 frame out of every 32 frames).

In order to proceed with Image feeding optimisation, we initially process one frame for every p frames; and as per the Algorithm 1, at the beginning $p = q = 32$. If the system detects any kind of HAZMAT signs, we decrease p by dividing it by 2 (i.e. $p = p \div 2$) and again we process one frame per p frames. As long as we repeatedly see HAZMAT signs in the input frames we halve the p down to 16 then 8, 4, 2, and 1. Therefore, for every new time, t, we process a double number of frames than the previous time, $t - 1$. As long as we see the HAZMAT signs, p will keep decreasing until it reaches to 1. Otherwise, in case of no HAZMAT sign detection, p will be doubled repeatedly until its maximum possible value (i.e. 32 in the above example). In other words, we only let the system analyse more images frames, and consequently more CPU usage, if the system identifies a high chance of the HAZMAT signs in the current and upcoming frames.

Fig. 6 depicts the amount of CPU usage after applying the image feeding optimisation. The horizontal axis represents the *time* for 90 s, and the vertical axis represents the *percentage* of the CPU usage. The diagram shows the usage of 6 processor cores of an Intel Core i7 CPU, in different colours. As can be seen in Fig. 6, there are some cases that we have very limited CPU usage (under 20% in total). These are the instances where there has been no clue of the HAZMAT signs in the input images, and consequently a minimum number of frames has been assessed per second. On the one hand, there are three cases where the CPU usage has rapidly increased to above 82% in just a few seconds. These are the cases that we have detected some ROI as potential HAZMAT singes, and consequently more and more input frames have been analysed.

Algorithm 2: Adaptive Non-Maximal Suppression

```
Function ANMS(B, S, C, t):
    D ← {}
    while B ≠ empty do
        m ← selectMaximumConfidence(S, C)
        M ← b_m
        D ← D ∪ M
        B ← B − M
        for b_i in B do
            if IoU(M, b_i) ≥ t then
                B ← B − b_i
                S ← S − s_i
                C ← C − c_i
            end
        end
    end
    return D, S, C
End Function
```

As per Fig. 6, the average CPU core usage for a 90-second sample video is around 40% while without input feeding optimisation, the CPU usage would be constantly around 82%.

5.3. Adaptive non-maximal suppression

Traditional object detection algorithms use a multi-scale sliding-window-based approach to search for a particular object in a window. Each window receives a score, depending on the number of matching features found inside the query window. Windows with a higher score than the set threshold will be marked as the candidate object regions. The final step of such approaches is to remove multiple neighbouring bounding boxes which point to the same instance of the object. This post-processing step is called non-maximal suppression (NMS).

In DNN-based object detection algorithms, the sliding-window approach is replaced with category independent region proposals using a CNN. Similarly, a non-maximal suppression is also used in DNN based models to obtain the final set of detections. This significantly reduces the number of false positives (Bodla et al., 2017).

Although occluded or overlapped signs may rarely appear in rescue operation scenes, we have to be cautious about it before suppressing them with a blind NMS approach. Conventional NMS models significantly suppress the overlapped bounding boxes, by keeping only the most confident ones and skipping the less confident bounding boxes. Non-Maximal Suppression also ensures that we would not have any redundant or extraneous bounding boxes. In some cases, the YOLO can detect partially overlapped objects and signs; however, it does not apply non-maximal suppression. Therefore, we would require to explicitly apply the NMS in our model. The standard NMS implementations (e.g. in OpenCV) does not care about the class of the occluded signs, and simply suppresses them all together. In contrast, we implement an adaptive version of Non-maximum suppression functions which we call it ANMS. The ANMS not only takes the class of the bounding boxes into account but also considers the confidence score of each bounding box to maintain the maximum benefit of the NMS without suppressing the important information.

As per the Algorithm 2, the suppression process depends on a threshold value and the selection of threshold value is a key parameter in the performance of the model. As shown in the algorithm, instead of selecting the highest confidence value of a set of neighbouring bounding boxes, we select the highest confidence value of the same classes to make sure we do not suppress different classes even with lower confidence levels.

In Algorithm 2, B_i is the list of initially detected bounding boxes, D_j is the list of final detections, S_i represents the corresponding detection

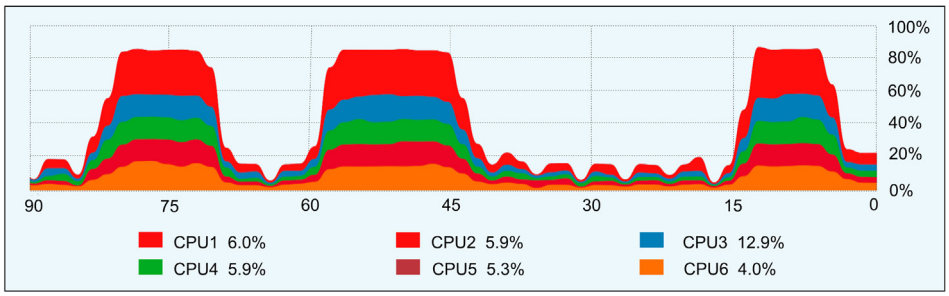


Fig. 6. A 90-Second sample graph of the 6-Core CPU usage for the proposed Image feeding optimisation.

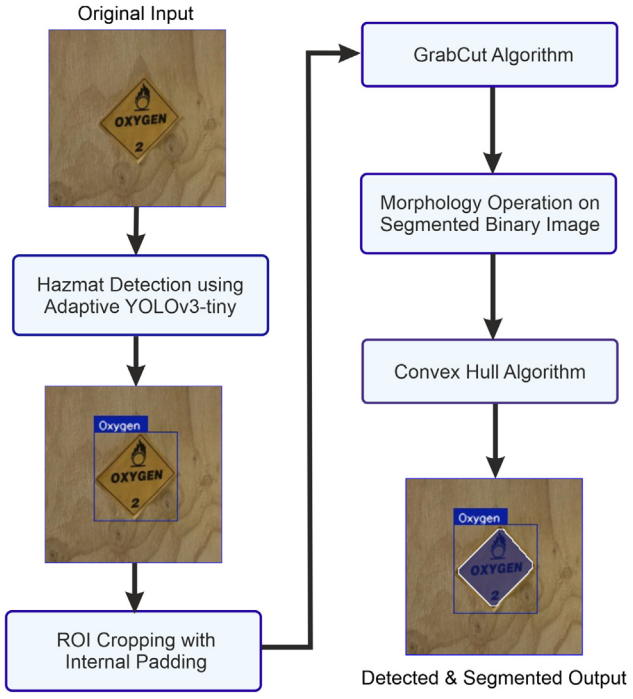


Fig. 7. Flowchart of the proposed method.

scores, C_k contains corresponding detection classes, and t is the NMS threshold. Non-maximum suppression starts with a list of detection boxes B with scores S and classes C. After choosing the bounding box with the maximum score m in the class C, the remaining bounding boxes from the same class will be removed from the set B and then appends it to the D. It also removes any overlapped boxes with M in the set B with a threshold greater than t. This process is repeated for the remaining boxes B and classes.

The NMS algorithm that we use in ANMS is based on the Blazing Fast-NMS developed by Tomasz Malisiewicz (Malisiewicz, 2011) which is over 100x faster than older NMS algorithms.

5.4. GrabCut segmentation

YOLO bounding box outputs are in the form of upright rectangles; however, the predicted HAZMAT signs may have been rotated. This may cause part of the background segments to also appears in the HAZMAT sign bounding boxes (see Fig. 7).

To cope with this, background removal methods can be used to separate the detected HAZMAT sign from the background. Such operations help to have a more accurate position of the object and its approximate angle. In some environments, the HAZMAT sign may have a colour match with the background or other objects next to the detected sign, and the algorithm may consider that these objects and colours are part of the sign, and ultimately make a mistake in separating the sign from the background. To solve this problem, it is recommended to use edge detection and noise elimination algorithms.

GrabCut (Rother et al., 2004) is an image segmentation method, based on iterative graph cuts as in Chen et al. (2019), Gulshan et al. (2011), Jeyalakshmi and Radha (2019), Li et al. (2018). The algorithm estimates the colour distribution of the target object and that of the background using a Gaussian mixture model. In GrabCut based segmentation algorithm, we pass our region of interest in form of a bounding box to extract the foreground (i.e. the HAZMAT sign) from the background. Since our Neural network already provides a rather accurate bounding box for the HAZMAT region, and the HAZMAT signs are colour-coded signs, we believe this will be an appropriate approach to take the advantage of using an adapted GrabCut technique for a dimension-independent and high-resolution segmentation.

As one of the very recent research works, but in a different application, Ünver and Ayan (Ünver & Ayan, 2019) also use the GrabCut technique for lesion skin segmentation. To the best of our knowledge, no research has been performed on the utilisation and adaptation of the GrabCut technique for HAZMAT sign segmentation. In the next section (experimental results) we demonstrate a very high rate of Intersection over Union (IoU) achieved, in comparison with the ground truth data and other conventional metrics.

We pass the YOLO bounding boxes outputs to the adapted GrabCut algorithm by applying a small ($\cong 5\%$) internal padding. Fig. 7 illustrates a flowchart of the segmentation method. The final result includes two types of pixel-wise segments: HAZMAT and non-HAZMAT segments.

After that, we use Convex Hull to gain a more accurate polygon segmentation that encompasses the HAZMAT sign boundary (See Fig. 8). The convex hull for a set of pixel points S in n dimensions is the intersection of all convex sets containing S. For N pixel points $p_i = p_1, \ldots, p_N$, the convex hull C is given by the following expression:

$$C = \left[\sum_{j=1}^{N} \lambda_j p_j \; : \; \lambda_j >= 0 \quad \text{for all } j \quad \text{and} \quad \sum_{j=1}^{N} \lambda_j = 1 \right] \tag{1}$$

were $\lambda_j = l_i / L$, and l_i is the ratio of the length of each convex edge i to n to the total length of all edges $L = \sum_{j=1}^{n} l_j$.

5.5. Data augmentation

Collection and preparation of a large and multi-faceted dataset have always been a challenge in training deep-learning based models. Having a larger dataset leads to a better training and higher accuracy. We have created an event logger for the detection service that captures and saves the HAZMAT images during the real-world operations of the rescue robot, to create a secondary HAZMAT dataset. The collected HAZMAT signs can either be feed forwarded to the network for further training or to be saved and annotated later by an expert to expand the main dataset. This creates a more comprehensive train set for further development of the model, hence, more accurate rescue operations in the future.

Fig. 8. Overall view of HAZMAT detection and segmentation, using our adaptive YOLOv3-tiny and GrabCut Segmentation.

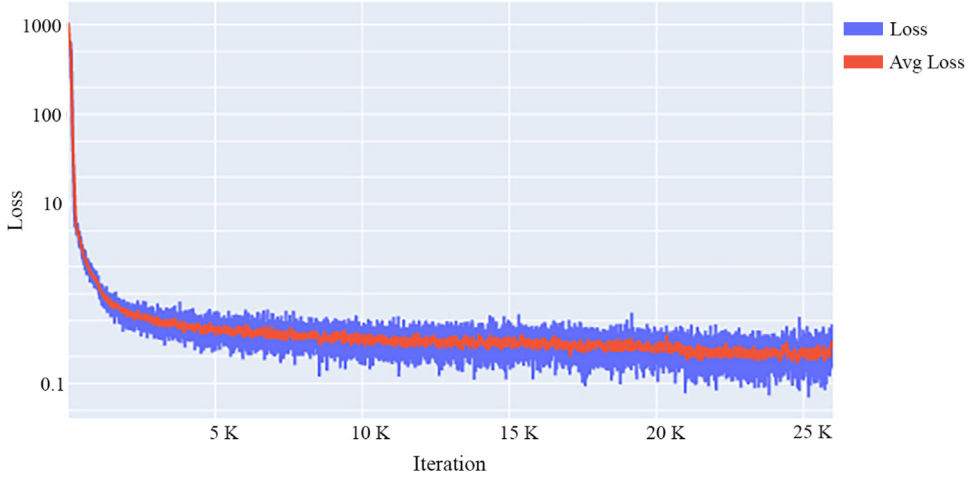


Fig. 9. The training graph (Loss and Average Loss improvements over time and iterations).

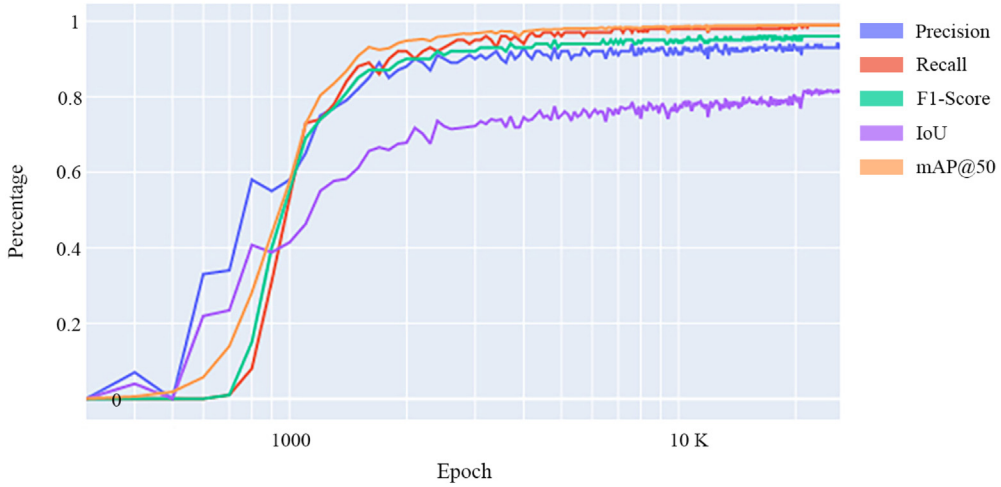


Fig. 10. The training graph (Metrics vs. iteration).

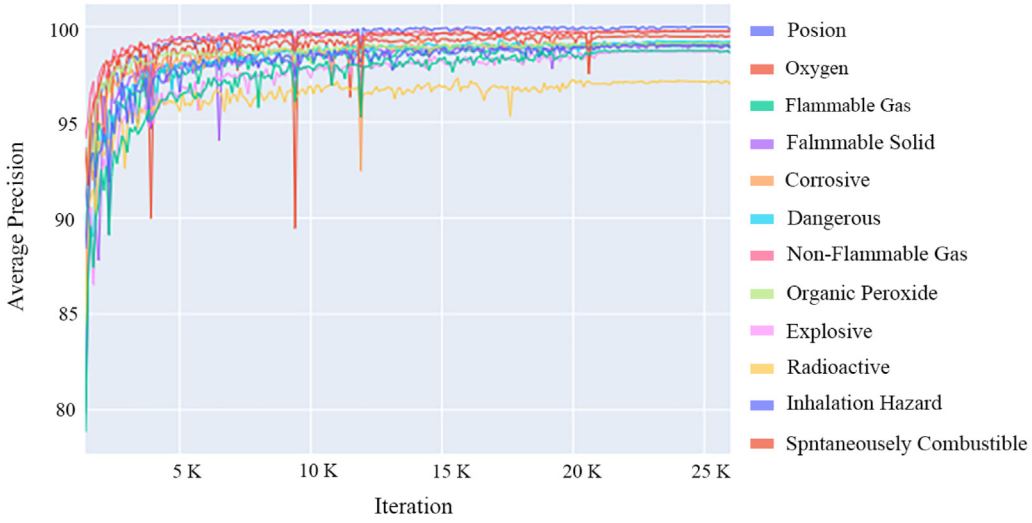


Fig. 11. The training graph (Each Class AP vs. iteration).

Table 2
Training configuration of the YOLOv3-tiny for our HAZMAT detection robot.

Parameter	Value
Batch Size	64
Subdivisions	16
Momentum	0.9
Decay	0.0005
Burn In	1000
Learning Rate	0.001
Max Batches	26 000

6. Experimental results

We trained our developed model on a PC platform, equipped with an Intel Core i7-6700 CPU, NVIDIA GeForce GTX 1080Ti GPU, 8 GB of Memory, and Ubuntu 18.04 OS.

To configure our custom YOLO model, we had to consider a trade-off between speed and accuracy. After several experiments, we find the best size of the input image for our model as 576 × 576. Table 2 provides the details of our custom training setting for the YOLOv3-tiny deep neural network. We set the learning rate as 0.001 with a batch size of 64. In the final stages of iterations in the training phase (between the iteration numbers 20 800 and 23 400), we multiplied the learning rate to 0.1 to make it smaller and proceed with a more precise weight adjustments, to prevent overfitting. Fig. 9 shows the loss and average loss value of the model during the training phase, and after $25K$ iterations.

6.1. Evaluation metrics

In order to assess the robustness of our method, we conducted five different evaluation metrics to analyse the average precision of each class as follows:

- **Precision rate:**
$$\frac{\sum TruePositives}{\sum(TruePositives + FalsePositives)}$$
- **Recall rate:**
$$\frac{\sum TruePositives}{\sum(TruePositives + FalseNegatives)}$$
- **F1-score:**
$$2\times\left(\frac{Precision\times Recall}{Precision + Recall}\right)$$
- **Average Intersection over Union:**
$$IoU = \frac{Summed\,AreaOfOverlap}{AreaOfUnion}$$
- **Average Precision:**
$$AP = \sum_{n}(R_n - R_{n-1})P_n$$

Using 80% of the dataset for the train phase and the rest of 20% unseen samples as the test dataset we achieved the following results: Precision rate = 93.54%, Recall-rate = 99.53%, F1-score = 95.98%, Average Intersection over Union (IoU) = 81.83%.

In order to provide further evaluations, we also considered the mean average precision (mAP) as another standard metric proposed in the Microsoft COCO article (Chen et al., 2015), where mean AP is the average precision over multiple Intersection over Union (IoU).

- **Mean Average Precision:**
$$mAP = \frac{\sum_{q=1}^{Q} AP(q)}{Q}$$

In our proposed neural network with an IOU threshold of 50%, we achieved the mean average precision (mAP@50) of 99.03%.

Fig. 10 shows the gradual improvement of the Precision rate, Recall rate, F1-score, Average IoU, and mAP after 25,000 Epochs. Fig. 11 demonstrates the average precision rate of the proposed DeepHAZMAT model for all HAZMAT signs discussed in Section 5.1.

Based on the visual appearance of the graphs shown in Figs. 9–11, it can be also confirmed that the model is not suffering from overfitting issue, so we can conclude that the hyperparameter tuning of the system has been successful and the system is robust and reliable enough in dealing with all of 13 discussed hazardous materials signs.

Looking at Figs. 12 and 13 we would like to reiterate the robustness and the performance of the system in two other challenging scenarios: HAZMAT sign detection in complex backgrounds and lighting conditions as well as detection of occluded or partially visible HAZMAT signs in complex real-world scenes and backgrounds, performed by ARKA robot.

Table 3 shows the performance of the model for every class using six evaluations metrics including Average Precision (AP), Precision Rate (PR), Recall Rate (RR), Accuracy (ACC), F1-Score, and Intersection over Union (IoU). Paying attention to the green and red numbers in each column of the table (as the best and weakest performances, respectively), it can be seen that while the algorithm is very robust in dealing with all 13 categories of HAZMAT signs, it performs slightly

Table 3
The experimental results and metrics for the accuracy of each HAZMAT class.

Class	AP %	PR %	RR %	ACC %	F1-score %	IoU %
Poison	99.96	97.62	100.00	99.80	98.78	89.32
Oxygen	99.78	97.63	98.76	99.72	98.19	88.10
Flammable	99.18	96.46	97.08	99.42	96.77	85.91
Flammable-solid	99.05	98.19	96.95	99.57	97.57	86.95
Corrosive	99.47	98.39	98.05	99.73	98.24	86.95
Dangerous	99.58	96.20	98.75	99.52	97.46	87.56
Non-flammable-gas	99.90	98.66	98.90	99.82	98.77	88.53
Organic-peroxide	99.34	98.51	98.51	99.80	98.51	88.84
Explosive	99.08	95.02	97.66	99.21	96.32	87.78
Radioactive	98.40	95.58	95.82	99.20	95.70	86.25
Inhalation-hazard	99.27	95.97	97.80	99.48	96.88	87.43
Spontaneously-combustible	99.64	97.99	99.34	99.80	98.66	88.36
Infectious-substance	99.20	95.85	97.55	99.47	96.69	87.50
All class metrics average	**99.37**	**97.09**	**98.09**	**99.58**	**97.58**	**87.65**

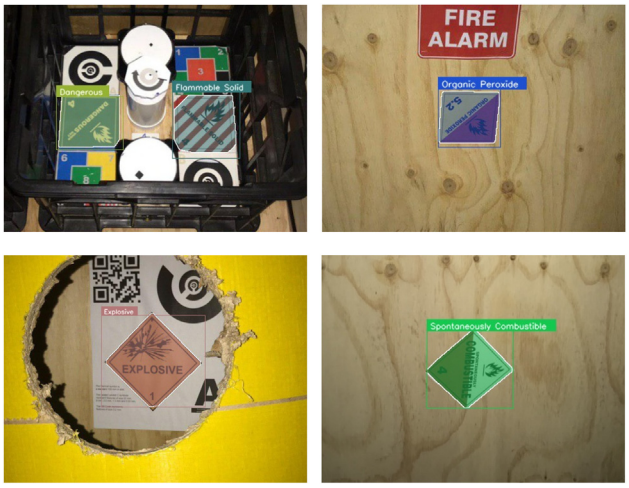


Fig. 12. Successful detection of HAZMAT signs by ARKA robot on a challenging test field in Sydney, Australia.

Fig. 13. Examples of signs detection and segmentation in challenging lighting conditions with significant occlusions.

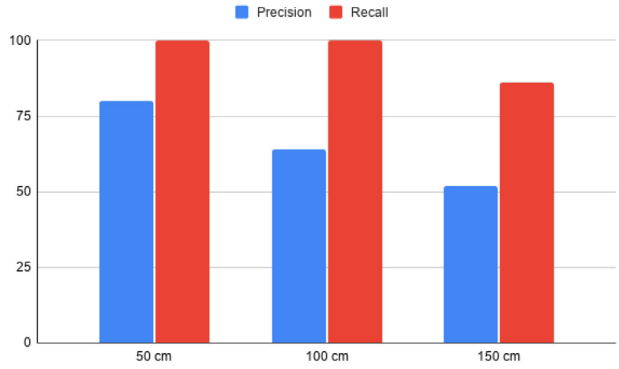


Fig. 14. Precision and recall rates for over different distances to HAZMAT signs.

better for the Poison HAZMAT sign detection and slightly weaker in Radioactive sign detection. This means we probably need to add more diversity of sample radioactive HAZMAT signs to the training set. This will lead to gain a more balanced performance for all signs.

We also conducted further experiments to assess the effect of the camera distance to the HAZMAT signs on the model precision and recall rate. A second dataset with 50 sample HAZMAT signs was created in 3 sub-categories with the distances of 50 cm, 100 cm, and 150 cm. The proposed model was evaluated on this dataset and the results are shown in Fig. 14. While the recall rate did not change significantly for the distances of up to 100 cm, the precision rate was decreasing for farther distances.

Fig. 15 provides a normalised confusion matrix to visualise the accuracy of the proposed method for every class. The horizontal axis demonstrates the actual labels and the vertical axis shows the predicted labels. The confusion matrix shows there are very limited instances where the DeepHAZMAT model may confuse the explosive and radioactive signs, interchangeably. A similar misclassification can be seen for Non-flammable gas signs, otherwise we can see a nearly perfect classification results based on the matrix diagonal.

In Table 4 we compare the performance of the proposed DeepHAZMAT methodology with five other models (one classic model and four state-of-the-art DNN-based models). We evaluated each method against nine metrics including five accuracy related metrics, three versatility based features, and finally, the overall speed of the model.

As the table represents, none of the evaluated models is a definite winner in all metrics; However, DeepHAZMAT performs as the top in five major metrics including recall rate, speed, segmentation, the ability of multiple HAZMAT sign detection, and adaptive bounding box feature. The proposed model also stays as the second best in terms of other accuracy metrics.

The proposed DeepHAZMAT system only performs less than 1% weaker in non-winning metrics which is negligible compared to other important outperforming features such as speed and extra capabilities of the model. Considering the main objectives of this research, which was accurate multiple HAZMAT sign detection in challenging lighting conditions and environments, with restricted computational resources, the model well supports our requirements. SIFT performs as the weakest, and EfficientDet-D1 cannot outperform the YOLO family in most cases. It also takes more computational resources.

6.2. Limitations and challenges

We identified a set of limitations, challenges, and research gaps that can be considered in future studies.

Implementing a very accurate segmentation operation prevents us from a real-time processing. This is because there is an additional operation post the detection phase. If these two operations could be merged, we may have better a output in real-time processing.

DeepHAZMAT cannot tell the direction or angle of HAZMAT signs. Angle detection requires an accurate segmentation operation followed by a cardinal point detection.

In segmentation operations, it might be difficult to differentiate a sign from its background with the same colour.

Since we are not considering any text processing in our algorithm, we cannot separate HAZMAT signs of the same class with different

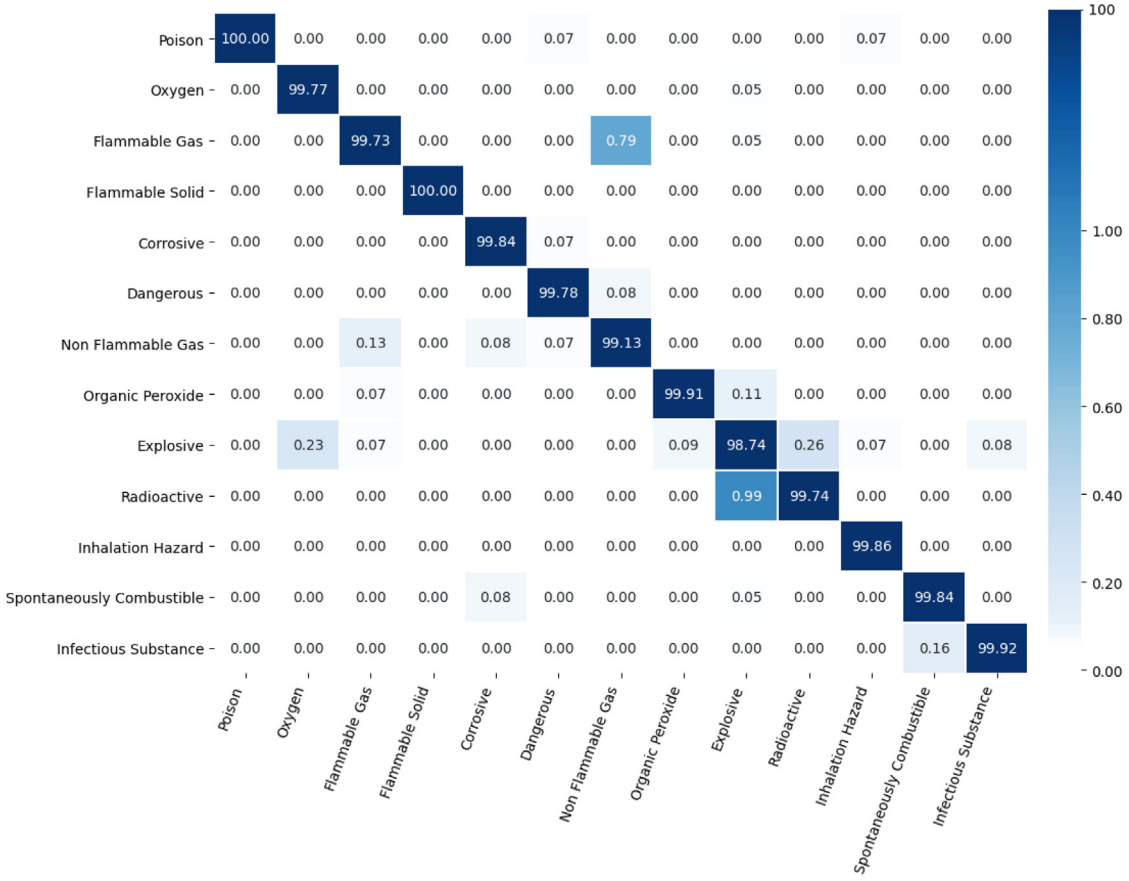


Fig. 15. Confusion Matrix of the 13 HAZMAT Signs Based on the Precision Rate.

Table 4
Performance Comparison of the proposed method with a common conventional method as well as other deep neural networks based HAZMAT detection techniques.

Metrics/Methodology	SIFT	YOLOv2	YOLOv3	YOLOv4	EfficientDet-D1	Proposed method
Average recall rate	26.42%	99.23%	99.21%	99.13%	93.23%	99.53%
Average precision rate	64.05%	91.86%	94.98%	92.00%	90.91%	93.54%
Average IOU rate	75.98%	80.86%	85.75%	81.46%	87.48%	81.83%
mAP rate	64.05%	99.70%	99.37%	99.28%	98.78%	99.03%
F1-score	33.80%	95.40%	97.05%	96.00%	95.92%	95.98%
Adaptive bounding box	No	No	No	No	No	Yes
Object detection	Single	Multiple	Multiple	Multiple	Multiple	Multiple
Segmentation	No	No	No	No	No	Yes
Overall speed	4 fps	2 fps	5 fps	5 fps	2 fps	32 fps

letters or numbers. For example, some HAZMAT signs have a number that cannot be identified with the proposed algorithm. Fig. 16 shows samples of similar HAZMAT signs with different letters and texts.

Based on our study, the performance of the model will be affected at a distance of more than 1 metre (Fig. 17 right column).

Despite the aforementioned challenges, in RoboCup competitions usually the environment and the field is free of unusual and strange signs. Therefore, the proposed model can efficiently recognise the HAZMAT signs, in real-time, and also with a higher average performance comparing to the state-of-the-art in HAZMAT sign recognition.

7. Conclusions

In this paper we presented a robust Computer Vision and Machine Learning based Robotic system that can localise, classify, and segment HAZMAT signs in hazardous rescue fields. The proposed methodology enabled us to confidently detect the presence of hazardous materials signs, regardless of the particular lighting situation, over a wide range of distances and under varying degrees of rotation. The trained model is also able to detect occluded, overlapped, and partially visible signs. The experimental results showed the DeepHAZMAT model is more accurate and faster than some other common and state-or-the-art research works such as Cai et al. (2020), Edlinger et al. (2019), Lowe (2004a) and Tijtgat et al. (2017). The developed DNN-based system was fast enough to be implemented in Mobile robots, using a single Intel NUC Core i7 embedded system for robust and real-time hazard label detection, recognition, identification, localisation, and segmentation, thanks to skipping redundant input data loads as well as adaptation of the YOLOv3-tiny for our real-time robotics application. As possible future work we suggest developing an optical character recognition method for text recognition inside the HAZMAT signs and to detect whole HAZMAT signs without selecting the background areas. In the

Fig. 16. Examples HAZMAT signs in the same class, but with different texts.

Fig. 17. Detection/Segmentation results in various distances.

interest of reproducible research, we have publicly released our unique HAZMAT-13 dataset as well as the implemented code in our GitHub DeepHAZMAT repository for the benefit of other researchers in the field. To the best of our knowledge, we are the first that publish such a large and comprehensive dataset of HAZMAT signs with the ground truth annotations, to the rescue robotics community.

References

Anderson, R. (2018). Inland Northwest Health Services (INHS).

Bodla, N., Singh, B., Chellappa, R., & Davis, L. (2017). Soft-NMS – Improving object detection with one line of code. In *ICCV*. http://dx.doi.org/10.1109/ICCV.2017.593.

Cai, J., Hou, J., Lu, Y., Chen, H., Kneip, L., & Schwertfeger, S. (2020). Improving CNN-based planar object detection with geometric prior knowledge. In *2020 IEEE international symposium on safety, security, and rescue robotics (SSRR)* (pp. 387–393). http://dx.doi.org/10.1109/SSRR50563.2020.9292601.

Chen, X., Fang, H., Lin, T.-Y., Vedantam, R., Gupta, S., Dollár, P., & Zitnick, C. L. (2015). Microsoft coco captions: Data collection and evaluation server. arxiv preprint arXiv:1504.00325.

Chen, Q., Xu, Q., Li, L., & Luo, B. (2019). Optimal bounding box and GrabCut for weakly supervised segmentation. In *2019 3rd international conference on electronic information technology and computer engineering (EITCE)* (pp. 1765–1770). http://dx.doi.org/10.1109/EITCE47263.2019.9094973.

Edlinger, R., Zauner, M., & Zauner, G. (2019). Hazmat label recognition and localization for rescue robots in disaster scenarios. In *IS and T international symposium on electronic imaging science and technology*: *vol. 2019, IS&T electronic imaging 2019* (7th ed.). http://dx.doi.org/10.2352/ISSN.2470-1173.2019.7.IRIACV-463, Intelligent Robotics and Industrial Applications using Computer Vision 2019 ; Conference.

Gossow, D., Pellenz, J., & Paulus, D. (2008). Danger sign detection using color histograms and SURF matching. In *2008 IEEE international workshop on safety, security and rescue robotics* (pp. 13–18). IEEE, http://dx.doi.org/10.1109/SSRR.2008.4745870.

Gulshan, V., Lempitsky, V., & Zisserman, A. (2011). Humanising grabcut: Learning to segment humans using the kinect. In *2011 IEEE international conference on computer vision workshops (ICCV Workshops)* (pp. 1127–1133). IEEE, http://dx.doi.org/10.1109/ICCVW.2011.6130376.

Hirose, K. (2012). 2011 Fukushima Dai-ichi nuclear power plant accident: summary of regional radioactive deposition monitoring results. *Journal of Environmental Radioactivity*, *111*, 13–17. http://dx.doi.org/10.1016/j.jenvrad.2011.09.003.

Hughes, J., Gilday, K., Scimeca, L., Garg, S., & Iida, F. (2019). Flexible, adaptive industrial assembly: driving innovation through competition: Flexible manufacturing. *Intelligent Service Robotics*, http://dx.doi.org/10.1007/s11370-019-00292-9.

Itti, L., Koch, C., & Niebur, E. (1998). A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *20*(11), 1254–1259. http://dx.doi.org/10.1109/34.730558.

Jeyalakshmi, S., & Radha, R. (2019). A novel approach to segment leaf region from plant leaf image using automatic enhanced GrabCut algorithm. *Compusoft*, http://dx.doi.org/10.1186/S13007-017-0245-8.

Kim, M., & Suh, I. (2019). Active object search in an unknown large-scale environment using commonsense knowledge and spatial relations. *Intelligent Service Robotics*, *12*, http://dx.doi.org/10.1007/s11370-019-00288-5.

Li, Y., Zhang, J., Gao, P., Jiang, L., & Chen, M. (2018). GrabCut image segmentation based on image region. In *2018 IEEE 3rd international conference on image, vision and computing (ICIVC)* (pp. 311–315). IEEE, http://dx.doi.org/10.1109/ICIVC.2018.8492818.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, *60*(2), 91–110. http://dx.doi.org/10.1023/B:VISI.0000029664.99615.94.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, *60*(2), 91–110. http://dx.doi.org/10.1023/B:VISI.0000029664.99615.94.

Loy, G., & Barnes, N. (2004). Fast shape-based road sign detection for a driver assistance system. In *2004 IEEE/RSJ international conference on intelligent robots and systems (IROS)(IEEE Cat. No. 04CH37566), Vol. 1* (pp. 70–75). IEEE, http://dx.doi.org/10.1109/IROS.2004.1389331.

Maldonado-Bascon, S., Lafuente-Arroyo, S., Gil-Jimenez, P., Gomez-Moreno, H., & Lopez-Ferreras, F. (2007). Road-sign detection and recognition based on support vector machines. *IEEE Transactions on Intelligent Transportation Systems*, *8*(2), 264–278. http://dx.doi.org/10.1109/TITS.2007.895311.

Malisiewicz, T. (2011). *Blazing fast NMS*. Massachusetts Institute of Technology.

Najafi, F., Bagheri, H., Bonakdar Hashemi, N., Pouryayvali, A., Ahadi, M., Mohammad, S., Mojahedpour, M., & Sharifi, A. (2019). RoboCup rescue 2019 team description paper MRL. http://dx.doi.org/10.13140/RG.2.2.30334.33601/1.

Ohno, K., Kawatsuma, S., Okada, T., Takeuchi, E., Higashi, K., & Tadokoro, S. (2011). Robotic control vehicle for measuring radiation in fukushima daiichi nuclear power plant. In *2011 IEEE international symposium on safety, security, and rescue robotics* (pp. 38–43). IEEE, http://dx.doi.org/10.1109/SSRR.2011.6106792.

Pao, D., Li, H., & Jayakumar, R. (1992). Shapes recognition using the straight line hough transform: theory and generalization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *14*(11), 1076–1089. http://dx.doi.org/10.1109/34.166622.

Parra, A., Zhao, B., Haddad, A., Boutin, M., & Delp, E. J. (2013a). Hazardous material sign detection and recognition. In *2013 IEEE international conference on image processing* (pp. 2640–2644). IEEE, http://dx.doi.org/10.1109/ICIP.2013.6738544.

Parra, A., Zhao, B., Haddad, A., Boutin, M., & Delp, E. J. (2013b). Hazardous material sign detection and recognition. In *2013 IEEE international conference on image processing* (pp. 2640–2644). http://dx.doi.org/10.1109/ICIP.2013.6738544.

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *2016 IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 779–788). http://dx.doi.org/10.1109/CVPR.2016.91.

Redmon, J., & Farhadi, A. (2018). YOLOv3: An incremental improvement. arxiv preprint arXiv:1804.02767.

Registei, T. F. (2012). NPRM, Notice of Proposed Rulemaking.

Rezaei, M., & Klette, R. (2017). *Computer vision for driver assistance: Simultaneous traffic and driver monitoring*. Springer.

Rezaei, M., Sarshar, M., & Sanaatiyan, M. M. (2010). Toward next generation of driver assistance systems: A multimodal sensor-based platform. In *The 2nd international conference on computer and automation engineering (ICCAE), Vol. 4* (pp. 62–67). IEEE, http://dx.doi.org/10.1109/ICCAE.2010.5451782.

Rezaei, M., & Shahidi, M. (2020). Zero-shot learning and its applications from autonomous vehicles to COVID-19 diagnosis: A review. *Intelligence-Based Medicine*, *3–4*, Article 100005. http://dx.doi.org/10.1016/j.ibmed.2020.100005.

RoboCup Comittee (2020). *RoboCup rescue rulebook*. RoboCup.

Rother, C., Kolmogorov, V., & Blake, A. (2004). "GrabCut" interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics*, *23*(3), 309–314. http://dx.doi.org/10.1145/1015706.1015720.

Sabzevari, R., Masoumzadeh, S., & Ghahroudi, M. R. (2008). Employing ANFIS for object detection in Robo-Pong. In *IC-AI* (pp. 707–712).

Tijtgat, N., Volckaert, B., & De Turck, F. (2017). Real-time hazard symbol detection and localization using UAV imagery. In *2017 IEEE 86th vehicular technology conference (VTC-Fall)* (pp. 1–5). IEEE, http://dx.doi.org/10.1109/VTCFall.2017.8288259.

Tzutalin (2020). Tzutalin/labelImg. URL: https://github.com/tzutalin/labelImg.

Ünver, H., & Ayan, E. (2019). Skin lesion segmentation in dermoscopic images with combination of YOLO and GrabCut algorithm. *Diagnostics*, *9*, 72. http://dx.doi.org/10.3390/diagnostics9030072.

Won, W.-J., Lee, M., & Son, J.-W. (2018). Implementation of road traffic signs detection based on saliency map model. In *2008 IEEE intelligent vehicles symposium* (pp. 542–547). IEEE, http://dx.doi.org/10.1109/IVS.2008.4621144.