Received: 9 May 2023 | *Revised:* 21 July 2023 | *Accepted:* 22 July 2023
DOI: 10.37917/ijeee.20.1.14 *Vol. 20 | Issue 1 | June 2024*

Iraqi Journal for Electrical and Electronic Engineering
Original Article

Open Access
IJEEE University of Basrah College of Engineering

Internet of Things Based Oil Pipeline Spill Detection System Using Deep Learning and LAB Colour Algorithm

Muhammad H. Obaid *[1], Ali H. Hamad [2]
[1]Informatics Institute for Postgraduate Studies, Iraqi Commission for Computers and Informatics, Baghdad,Iraq
[2]Department of Information and Communication Engineering, University of Baghdad, Baghdad, Iraq

Correspondance
*Muhammad H. Obaid
Informatics Institute for Postgraduate Studies,
Iraqi Commission for Computers and Informatics,
Baghdad, Iraq
Email: ms202110670@iips.icci.edu.iq

Abstract
Given the role that pipelines play in transporting crude oil, which is considered the basis of the global economy and across different environments, hundreds of studies revolve around providing the necessary protection for it. Various technologies have been employed in this pursuit, differing in terms of cost, reliability, and efficiency, among other factors. Computer vision has emerged as a prominent technique in this field, albeit requiring a robust image-processing algorithm for spill detection. This study employs image segmentation techniques to enable the computer to interpret visual information and images effectively. The research focuses on detecting spills in oil pipes caused by leakage, utilizing images captured by a drone equipped with a Raspberry Pi and Pi camera. These images, along with their global positioning system (GPS) location, are transmitted to the base station using the message queuing telemetry transport Internet of Things (MQTT IoT) protocol. At the base station, deep learning techniques, specifically Holistically-Nested Edge Detection (HED) and extreme inception (Xception) networks, are employed for image processing to identify contours. The proposed algorithm can detect multiple contours in the images. To pinpoint a contour with a black color, representative of an oil spill, the CIELAB color space (LAB) algorithm effectively removes shadow effects. If a contour is detected, its area and perimeter are calculated to determine whether it exceeds a certain threshold. The effectiveness of the proposed system was tested on Iraqi oil pipeline systems, demonstrating its capability to detect spills of different sizes.

Keywords
Dense Extreme Inception Network for Edge Detection(DexiNed), oil spill, MQTT Protocol, Xception networks, Holistically-Nested Edge Detection, LAB color space, and Drone.

I. Introduction

Pipeline monitoring plays a crucial role in early leak detection or prediction, which is vital for minimizing oil spill damage. Detecting a leak at an early stage facilitates timely repairs and ensures the proper functioning of the pipeline. In the past few decades, numerous methods have been introduced for leak detection in pipelines, each employing its own distinct operating principle and approach. These techniques can be broadly classified into three categories: external methods, visual or biological methods, and internal or computational methods [1].

The external technique involves the utilization of artificial sensing devices placed in close proximity to the monitored infrastructure for leak detection. Examples of such equipment include acoustic sensors, fiber optic sensing systems, vapor sampling devices, infrared thermography tools, and ground penetration radar systems. This approach offers several advantages and disadvantages. On the positive side, it is generally user-friendly and provides a rapid response time. However, it requires a highly skilled operator, involves direct contact with the leaking medium, incurs high execution costs, and may

suffer from a high false alarm rate [2]. The interior approach to leak detection makes use of internal fluid measurement instruments to monitor parameters such as flow rate, pressure, temperature, volume, density, and other factors that quantify the characteristics of the released products. These parameters are directly related to fluid flow in the pipelines and are monitored using software-based solutions that employ intelligent computational algorithms. Techniques like mass-volume balancing, digital signal processing, and pressure point analysis are examples of dynamic modeling methods. However, this approach has certain drawbacks, including sensitivity to leak size, potential false alarms, susceptibility to noise interference, high computational complexity, time-consuming implementation, and significant costs [3]. Leakage can also be detected through the biological method, which involves utilizing the sensory capabilities of trained dogs or human experts [4]. Leak detection is carried out using different artificial sensing devices located outside the pipelines, as part of the external method. Alternatively, leakage can be detected through the biological method, utilizing the sensory abilities of trained dogs or human experts. The interior approach for leak detection involves software-based solutions supported by sensors that monitor the internal environment of the pipeline, employing intelligent computational algorithms [5].

Drones are now a strong and trustworthy tool for professional data collecting. Uncrewed aerial vehicles (UAV) are now widely used in the private and public sectors, and their applications have spread to almost every industry, including the oil and gas sector. Drones provide a quicker, safer, and more cost-effective means of collecting massive amounts of data, which has the potential to completely alter the status quo of the petroleum industry's mapping, monitoring, inspection, and surveillance processes [6]. Computer vision plays a crucial role in the application of drones across various fields. It is utilized in controlling self-driving vehicles, performing tasks, navigation, and monitoring oil pipelines. Artificial intelligence algorithms, such as machine learning and deep learning, are employed to process images captured by drones and extract important features. These features are then utilized to make informed decisions regarding pipeline leakage [7]. Image segmentation is an essential step in computer vision projects, where different techniques are used to divide images into meaningful regions. This segmented output serves as a foundation for further image analysis and processing. While various segmentation methods exist, some requiring human interaction, it becomes impractical when dealing with a large number of images. In such cases, the LAB color space is commonly employed as it closely aligns with human vision, enabling automated image segmentation. This approach proves to be effective in achieving accurate and efficient segmentation results [8]. This work aims to monitor pipelines using a drone equipped with a Raspberry Pi, Pi camera, and GPS positioning system. The drone captures images and GPS information, which are transmitted to a base station via the MQTT IoT protocol through the cloud. The base station utilizes computer vision with a deep learning algorithm, specifically the DexiNed algorithm, to create thin edge maps of spills of varying sizes detected in the images, even in small areas. The LAB algorithm is employed to remove the effect of shadows and focus solely on the black color, which represents the natural color of spills. The circumference of the black spot is calculated using an algorithm that determines if it indicates a leak. In the event of a leak, the relevant information including the original image, spot size, and geographical location is sent in real-time via a web application to the competent authority. The remaining sections of the paper are laid out as follows. Firstly, the latest in related work is presented in the second section. Next, the theoretical foundations of the deep learning method used in this work are presented in the third section. Finally, the fourth section shows the design of the proposed system and the result obtained, while the working conclusions are presented in the fifth section.

II. Related Work

Researchers have dedicated their efforts to developing efficient methods for detecting leaks. Ibitoye et al. [9] introduced an innovative approach for detecting pipeline oil leaks by utilizing a convolutional neural network model to analyze real-time footage from Internet of Things (IoT) cameras installed along the pipelines. Li et al. [10] employed RGB images captured by optical cameras, which operate based on the red, green, and blue color systems, for decision-making in crude oil pipeline leakage detection. They utilized a convolutional neural network (CNN) to accurately classify different leakage rates by combining thermal and RGB images. Jiao et al. [11] suggested a combination of machine learning, drones, and traditional techniques for oil damage detection. They utilized deep convolutional neural network modeling to analyze oil spills and employed an updated Otsu approach to minimize false positive detection results. Krestenitis et al. [12] proposed the utilization of three deep-learning models, namely VGG16 (Visual Geometry Group 16), YOLOv3 (You Only Look Once, version 3), and Mask R-CNN (Region-based Convolutional Neural Networks), for the detection of oil spills in RGB photos. Alharam et al. [13] suggested the use of an unmanned device equipped with an onboard artificial intelligence processing system and a thermal camera to monitor oil spills. The system also incorporates an accurate classifier for efficient detection. Ravishankar et al. [14] identified issues in oil pipelines and proposed a solution called DARTS (Drone and Artificial Intelligence Reconsolidated Technological Solution), which combines drone technology with deep learning

techniques to monitor and predict the progression of these issues. Sharafutdinov et al. [15] proposed a comprehensive oil spill-detecting equipment for UAVs, including microwave radiometer sensor, radar, laser radar, infrared and ultraviolet spectrometers, which enable them to detect emergencies, analyze the extent and depth of an oil spill, and provide valuable images of geological strata composition. Mahdianpari et al. [16] developed a hierarchical object-based random forest (RF) technique to map soil contamination caused by pipeline breaches using high-resolution UAV images and electromagnetic induction (EM) surveying data, with salinity indexes used to identify soil pollution. Korlapati et al. [17] developed UAVs equipped with thermal infrared (IR) cameras for the detection of oil spills in ports, utilizing convolutional neural networks (CNN) and a low-power interference device. Shukla et al. [18] employed drones with non-contact sensors for external pipeline inspection, enabling automated inspection and tracking while flying low over the pipeline structure. The system uses onboard cameras for identification and tracking, eliminating the need for GPS data. Zhou et al. [19] developed vertical take-off and landing (VTOL) UAVs, which take off and land vertically, were deployed and communicated with to monitor pipelines. Low-power UAVs with directional antennas and long-range zoom cameras monitor pipeline sections in real-time when the pressure drops or third parties are active.

III. Edge Detection-Based Deep Learning

Among the many methods of image processing, edge detection is crucial. Data from the stunning accessory picture functions on edge as a set of grayscale pixels with a sharp transition. The field of edge detection has seen widespread implementation across a wide range of practical needs, from the medical to the industrial. It is possible to classify edge detection methods as either traditional edge detection or those based on deep learning [20]. DexiNed, an Extreme Inception Network for Edge Detection, is a cutting-edge technique for identifying edges with pinpoint precision. Holistically-Nested Edge Detection (HED) and Xception networks serve as inspirations. In addition to applying to every edge detection task without the requirement for training or fine-tuning, the method automatically generates narrow edge maps convincing to humans [21].

A. Dense extreme Inception network for edge detection

DexiNed is one of the CNN algorithms used for edge detection. It merges a reduction network (up-sampling block (UB)) with a dense Inception network (Dexi) to create DexiNed. Dexi creates feature maps from the input image and sends them to UB for training. While other deep learning (DL)-based edge detectors need to initialize the weight using pre-trained object recognition models, DexiNed is taught from scratch [22]. Fig. 1 shows the DexiNed architecture.

1) Dexi architecture

Dexi, which is based on the VGG16 architecture, comprises six major blocks inspired by the Xception network. The network utilizes an upsampling block to generate feature maps as outputs from its main blocks, and these feature maps are then used to produce intermediate edge maps. When all edge maps generated by the upsampling blocks are fed into the stack of learned filters at the end of the network, a merged edge map is created. Each of the six upsampling blocks has its own unique weights, and the blue blocks consist of a stack of two 3x3 kernel-size convolutional layers, batch normalization, and a rectified linear unit (ReLU) activation function. Fig. 1 illustrates how the max-pool is determined using the 3x3 kernels and stride 2. Similar to the multi-scale learning approach of HED, this design employs upsampling (indicated by the grey blocks in the diagram). Although DexiNed draws inspiration from Xception, the similarities end at the level of the framework's primary building blocks and interconnections [23].

2) Upsampling architecture

The proposed design implemented by DexiNed aims to enhance the quality of expected edge maps generated from captured images, particularly for oil slicks. The upsampling block plays a crucial role in thinning the edges. Fig. 2 illustrates the architecture, where the outputs from all DexiNed blocks are fed into the upsampling block, which consists of stacked conditional sub-blocks. Each building block consists of two layers: a convolutional layer and a deconvolutional layer. The first sub-block type is the simplest, while the second is the most complex. The Dexi outputs of the second sub-block serve as inputs to the first sub-block and are used when the discrepancy between the feature maps' scales and the ground truth (GT) is two. In the DexiNed architecture, when the variance exceeds two, the next sub-block is used, and this process continues until the GT feature map reaches a size of two. The GT feature map's scale is achieved by repeating this sub-block. The subsequent sub-block consists of a 1x1 kernel in the switching layer, followed by the ReLU activation function. The kernel size for deconvolution layers or convolutional switches is $s \times s$, where s represents the level scale of the input feature map. Each layer returns one filter, resulting in map features at the same scale as the GT size. The activation function is absent in the last transformation layer. Subblock 2 is similar to subblock 1, except that it has 16 filters instead of 1. Downsampling from subblocks can be performed using sub-pixel warping, warp transfer, and binary linear interpolation [24].

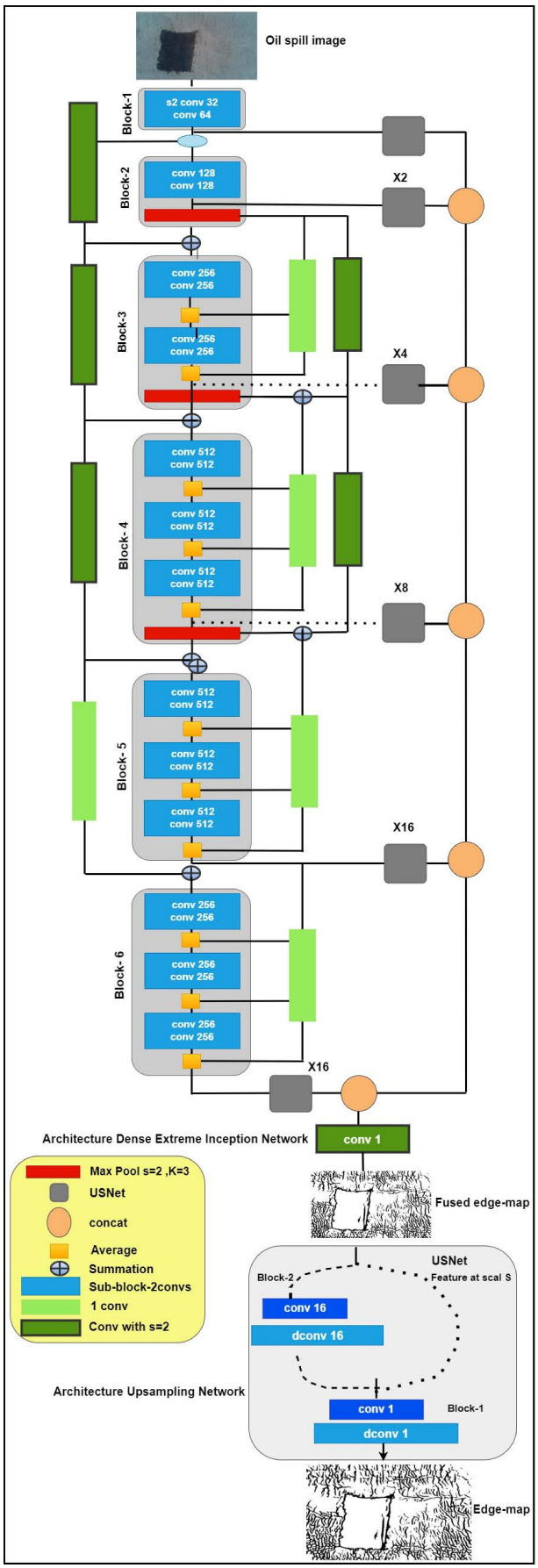


Fig. 1. DexiNed architecture [23]

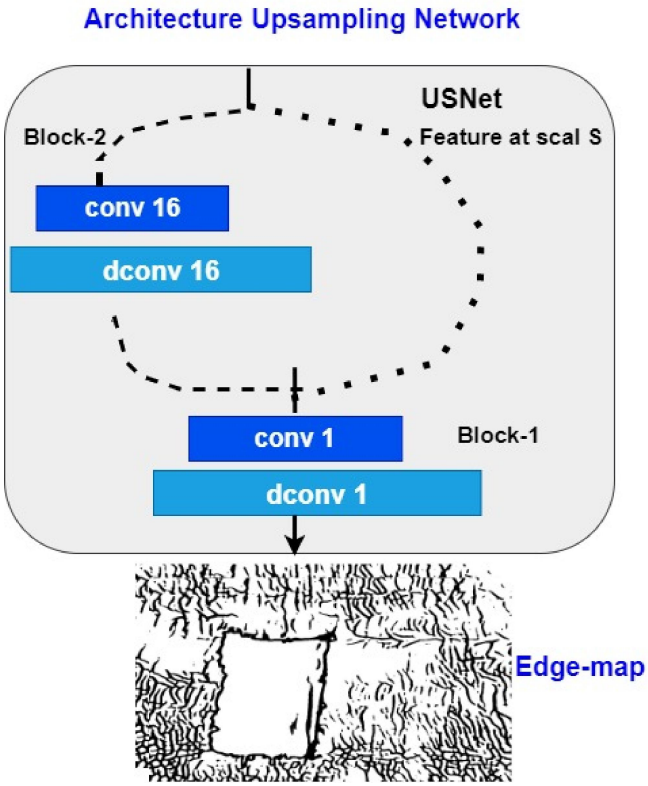


Fig. 2. Architecture upsampling network [24]

3) Loss functions

It is possible to express DexiNed in terms of a regression function ζ that is, $\widehat{Y} = \zeta(X,Y)$, where X is an input image, Y is the corresponding ground truth, and $\widehat{Y}$ It is a collection of predicted edge mappings. $\widehat{Y} = [\widehat{Y}_1, \widehat{Y}_2, ..., \widehat{Y}_N]$, where $\widehat{Y}_i$ has the same size as Y and is the number of outputs from each upsampling block (grey horizontal rectangles, Fig. 1); $\widehat{Y}_N$ is the result from the final fusion layer $f(\widehat{Y}_N = \widehat{Y}_f)$. Therefore, as this is a model, the same loss as (weighted cross-entropy) is used; hence, this problem is thoroughly supervised and addressed similarly.

$$\begin{aligned} i^{\mathbf{n}}(\mathbf{W}\cdot\mathbf{w}^{\mathbf{n}}) = -\beta \sum_{\mathbf{j}\in Y^+} \log\sigma\left(\mathbf{y_j 1} \mid \mathbf{X}; \mathbf{W}\cdot\mathbf{w}^{\mathbf{n}}\right) \\ -(\mathbf{1}-\beta) \sum_{\mathbf{j}\in Y^-} \log\sigma\left(\mathbf{y_j = 0} \mid \mathbf{X}; \mathbf{W}\cdot\mathbf{w}^{\mathbf{n}}\right). \end{aligned} \tag{1}$$

$$\mathscr{L}(\mathbf{W}\cdot\mathbf{w}) = \sum_{\mathbf{n=1}}^{\mathbf{N}} \delta^{\mathbf{n}} * \mathbf{l}^{\mathbf{n}}(\mathbf{W}\cdot\mathbf{w}^{\mathbf{n}}) \tag{2}$$

Each scale has its weight, denoted as W is the set of all network parameters, and w is the parameter with index, δ is a weight for each scale level. $\beta = |Y^-|/|Y^+ + Y^-|$ and

$(1-\beta) = |Y^+|/|Y^+ + Y^-|(|Y^-|, |Y^+|$ represent the edge and non-edge in the underlying truth, respectively) [25].

B. Extreme Inception networks(Xception)

The foundation of a convolutional neural network is a series of convolution layers that may be independently explored in detail [26]. The network has three different flows: the entrance flow, the middle flow (repeated four times with increasing filters at each iteration), and the exit flow. Fig. 3 illustrates Xception's network architecture. The entrance flow extracts coarse features through four convolutional layers, batch normalization, and the ReLU activation function. It is noteworthy that the number of filters doubles from the previous convolutional layers. In the intermediate flow, eight distinct convolutional layers extract more complex features, with each set of two layers doubling the filter count. The output flow extracts the most fine-grained characteristics using two independent convolutional layers. Subsequently, global average pooling is applied to reduce the 3x3 mapping to a 1x1 size. The softmax activation function is used for classifying oil spills [27].

C. Holistically-nested edge detection

The HED strategy is neural network based, with VGG16 as the main network node. It is a deep neural network-based edge detector for images. The algorithm takes a color picture as Input. It outputs a map of edges with confidence scores ranging from 0 to 1 (or, equivalently, 0 to 255) for the existence of edges in each pixel. The HED strategy bases itself on the VGG16 network's convolutional layer (encircled by a dotted box in Fig. 4). This section comprises 13 convolutional layers with 3×3 kernels and ReLU activation functions after each layer. The network consists of five subnetworks, each of which processes images at a different scale: the first subnet processes images at the full resolution of the input picture, while the others process images at half, quarter, eighth, and sixteenth that size. A max-pool operation with a 2×2 kernel and a 2×2 stride decreases the resolution while moving from one set to the next. The final result of HED combines the outputs of five VGG16 groups trained to represent edge maps at five distinct scales [28].

Two basic HED components are often used: multi-level feature learning, multi-scale prediction, and whole-image-based training. First, the HED network is a fully convolutional neural network that accepts the whole picture as input and outputs just the edge information. Regarding the second part, HED adjusts VGG16 by removing the last convolutional and fully connected layers and replacing them with intensive supervision. Then, they follow the convolutional layers with the lateral output layers. The greater the convolution kernel and the lower the side output, the deeper the network layer. The

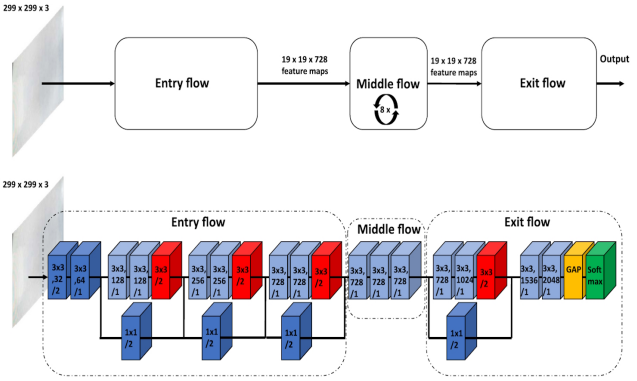


Fig. 3. Xception CNN architecture [27]

ultimate result is a fusion of many characteristics from the output's side branches [29].

1) Formulation

During training, the input training data set is represented as $S = (X_n, Y_n)n = 1, ..., N$ where sample $X_n = x_j^{(n)}, j = 1, ..., |X_n|$ represents the raw input image and $Y_n = y_j^{(n)}, j, ...|X_n|y_j^{(n)} \in 0, 1$ represents the matching ground truth binary edge map for the image X_n. Weights are represented as $w = w^{(1)}, ..., w^{(M)}$, and the network comprises M side-output layers. Computes the loss function at the picture level for auxiliary outputs. In the field of image-to-image training, as in shown Fig. 4 [30]: Can determine fusion layer loss function $\mathscr{L}$ By:

$$\mathscr{L}(W, w, h) = Dist\left(Y, \widehat{Y}_{fuse}\right) \tag{3}$$

To optimize for the minimum value of the following objective function using (backpropagation) accidental fall down a gradient by:

$$(W, w, h)^* = argmin\mathscr{L}(W, w) + \mathscr{L}(W, w, h) \tag{4}$$

Both the side output layers and the weighted-fusion layer's predictions on the edge map may be used in the testing step with image$X$ by:

$$\widehat{Y}_{fuse}, \widehat{Y}_{side}^{(1)}, ..., \widehat{Y}_{side}^{(M)} = CNN\left(X(W, w, h)^*\right) \tag{5}$$

These created edge maps may be aggregated further to provide a single result by:

$$\widehat{Y}_{HED} = Average\left(\widehat{Y}_{fuse}, \widehat{Y}_{side}^{(1)}, ..., \widehat{Y}_{side}^{(M)}\right) \tag{6}$$

IV. LAB Color Space

Shadows are a major challenge for many computer vision tasks, including segmentation, object identification, and counting. Therefore, many picture apps need users to identify and

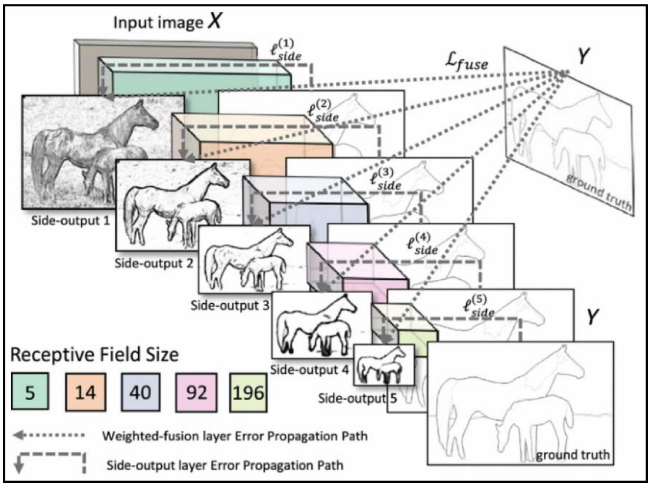


Fig. 4. Network architecture for HED [30].

eliminate shadows first. When an item blocks a light source's rays from penetrating a certain space, that space appears like a shadow. Even though shadows may provide important details about the things they cast onto. There are two sorts of shadows, hard and soft, and they differ in intensity. While the surface texture is still visible in soft shadows, harsh shadows are black and lack texture. Since other dark things may be viewed in place of shadows, accurate shadow detection is much more challenging. Also, many photos are required for camera calibration in many approaches. [31]. A luminance (lightness) channel, and two additional channels, A and B represent different chromaticity layers in this color space. Where a color lies on the red-green axis can be determined from the A* layer, and where it lies on the blue-yellow axis may be determined from the B* layer. The fact that this color space may be used to convey color information across multiple platforms and devices is its most notable characteristic. In Fig. 5 , the coordinates in the LAB* color space are clearly displayed. The brightness (L*) is represented along the middle vertical axis, with a range of values from 0 (complete darkness) to 100 (complete lightness). It is worth noting that A* cannot represent blue, yellow, green, or red since these colors are opposites on the coordinate axes. All axes in the color space have values ranging from positive to negative. Positive A* values indicate quantities of red, while negative values indicate green. Similarly, colors with positive B* values are more yellow, and those with negative B* values are bluer. The zero point on both axes represents grayscale neutrality. Only the brightness (L*) and color (R*, G*, B*) axes require values in this color mode, allowing users to independently adjust the image's luminance and hue. This unique feature enables the distinction between oil and non-oil spots in comparable photos. Fig. 6 shows the dimensions of a single RGB picture that may be processed by LAB, making it one of the essential technologies for identifying and eliminating shadows. Equation (8) is the result of linearly mapping the components of RGB space to the coordinate system.

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{bmatrix} 0.412453 & 0.357580 & 0.180423 \\ 0.212671 & 0.715160 & 0.072169 \\ 0.019334 & 0.119193 & 0.950227 \end{bmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix} \tag{7}$$

In the CIE-LAB color space, the X component represents the illumination, the Y component specifies the color, and the Z component is a normalized spectral ponderation curve based on statistical studies with human viewers. The color coordinates for each pixel in CIE-LAB are calculated through a non-linear modification of its X, Y, and Z values, using equations (9), (10), (11), and (12).

$$L^* = 116\left[f\left(\frac{Y}{Y_w}\right)\right] - 16 \tag{8}$$

$$a = 500\left[f\left(\frac{X}{X_w}\right) - f\left(\frac{Y}{Y_w}\right)\right] \tag{9}$$

$$b = 200\left[f\left(\frac{Y}{Y_w}\right) - f\left(\frac{Z}{Z_w}\right)\right] \tag{10}$$

$$f(x) = \begin{cases} t^{\frac{1}{3}} & \text{if } t > \left(\frac{6}{29}\right)^3 \\ \frac{1}{3}\left(\frac{29}{6}\right)^2 t + \frac{4}{29} & \text{if } t \leq \left(\frac{6}{29}\right)^3 \end{cases} \tag{11}$$

The Tristimulus of CIE-XYZ values around the "white spot" is denoted as Xw, Yw, and Zw.

$$\begin{pmatrix} X_w \\ Y_w \\ Z_w \end{pmatrix} = \begin{bmatrix} 0.9504 \\ 1.0000 \\ 1.0887 \end{bmatrix} \tag{12}$$

The initial mask of the color segmentation is obtained by computing the Euclidean distance using equation (14), with the help of the chromatic components a, b, and$L$, to determine the delta values [32].

$$\Delta E_{ab} = \sqrt{\Delta L^{*2} + \Delta a^2 + \Delta b^2} \tag{13}$$

V. MQTT Protocol

In the contemporary era, billions of interconnected smart gadgets and objects form a rapidly expanding network of linked

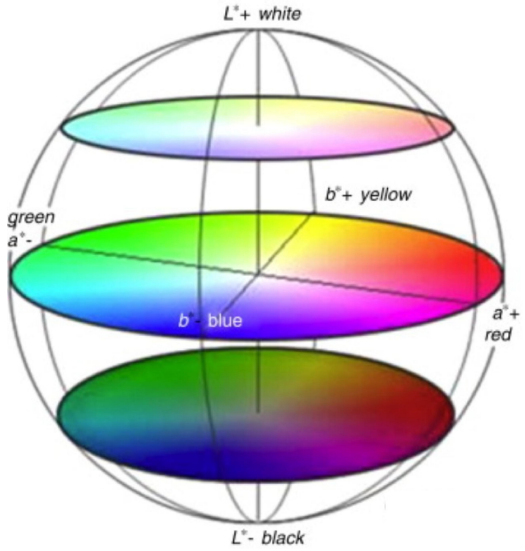


Fig. 5. RGB representation of the CIE-LAB color space [32].

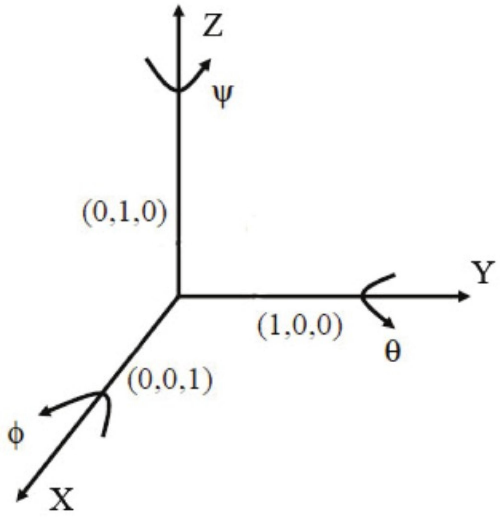


Fig. 6. Coordinate the system of a robot with six degrees of freedom [32] .

devices. These interconnected gadgets efficiently exchange information and services through sensor and event data transport protocols. Applications built upon these protocols enable seamless data collection, storage, processing, description, and analysis. Additionally, a key objective of the Internet of Things is to establish secure two-way communication among these connected devices [33] . The MQTT protocol is a free and open implementation of the Transmission Control Protocol (TCP) transport layer's publish-subscribe model. In the publish/subscribe model, clients can either subscribe to or publish on relevant subjects using a broker. When a client "publishes" a subject, any client nodes that have subscribed to it will receive the message. A node in the MQTT protocol can assume one of two roles: a publisher or a subscriber. The protocol was designed to maximize dependability and provide delivery assurance while minimizing the burden on networks and devices. Moreover, the protocol is well-suited for Machine-to-Machine (M2M) communications among IoT gadgets due to its consideration for limited resources like bandwidth and battery life [34].

VI. Proposed System

This work utilizes a drone to inspect pipelines while reducing human intervention by providing the drone with the coordinates of the pipeline paths. The drone is equipped with an embedded system, including a Raspberry Pi 4, Pi camera, and neo-6m GPS module. The drone is sent along the pipeline's path, which is 3m underground, in multiple rounds. The experiment was conducted on a 30 km segment of the Iraqi oil pipeline system. The embedded system captures images along the path at intervals of 1-2 meters. Table 1 illustrates the ground where the pipes extend, both before and after the spill, and demonstrates the implementation of the DexiNed algorithm on the images to compute the detected spill and its contour. Fig. 7 presents the schematic diagram of the proposed system. The image is then sent to the base station via the MQTT IoT protocol, along with its location provided by the GPS module. In the base station, the image is processed by the DexiNed algorithm to find different contours. Although there may be multiple contours in each image, only the ones with a significant area are considered potential spills. The LAB algorithm is then applied to identify contours in the image that have a black spot, which is indicative of a potential spill. Finally, another algorithm (Algorithm 1) calculates the area and perimeter of the detected spills. Fig. 8 illustrates the practical implementation of the proposed algorithm, and Table 2 presents a statistical analysis of the computed spill area in pixels and the spill diameter for the applied images. Additionally, Fig. 9 and 10 display the comparison results between the pixel area and perimeter of three spills at three different heights. During the first round of drone inspection, if the contour area of the spot is less than the threshold limit of $1m^2$, it is not considered a leak, and the image, along with its location, is kept in the database. In the second round of investigation, if the leakage oil area at the same location expands beyond the threshold limit as it is compared with the previous information stored in the database, it is considered a leak, and the information is transmitted. As shown in Fig. 11, the web applications page includes the old and new pictures of the leak and their location and size. But if the perimeter area of the spot since the first round of the drone is greater than the

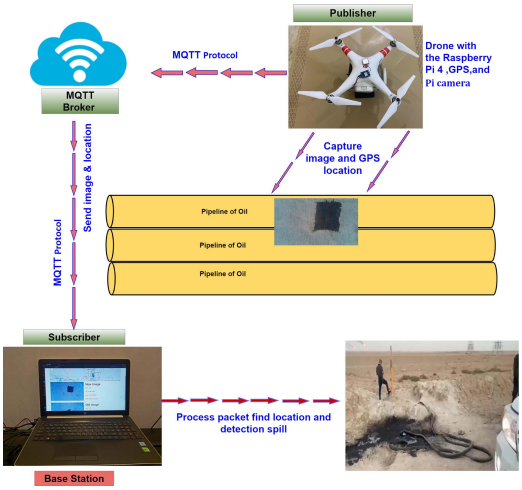


Fig. 7. Proposed system design and implementation.

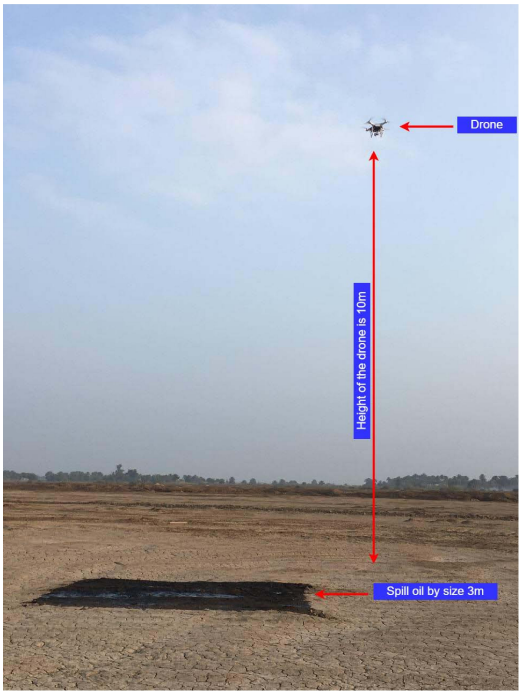
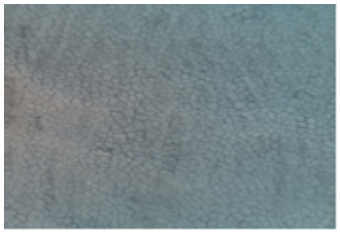


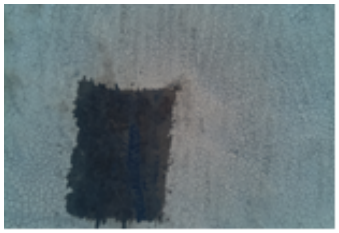
Fig. 8. Drone taking pictures of the oil leakage.

minimum, it is considered a direct leak, as shown in Fig. 12. By designing the proposed system and testing it on the Iraqi oil pipeline system, it has shown good results in detecting different sizes of spills and locating them in real-time. This has led to a reduction in costs and the devices and equipment used in this project.

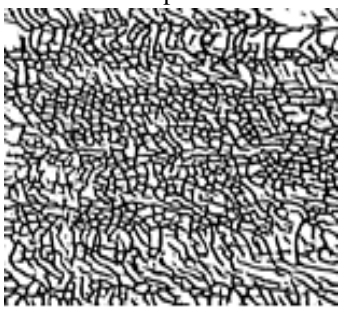
TABLE I.
$3M^2$ SPILL WITH 10 M HEIGHT BEFORE AND AFTER THE OIL SPILL



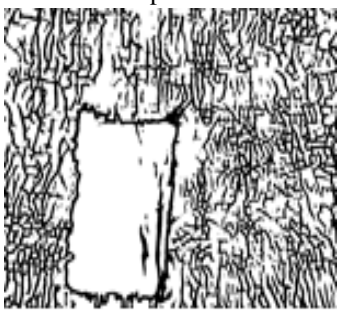
Original RGB image before the oil spill



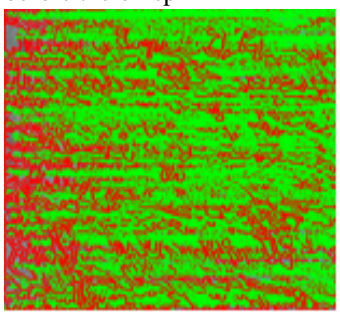
The original RGB image after the oil spill



Apply of algorithm DexiNed before the oil spill

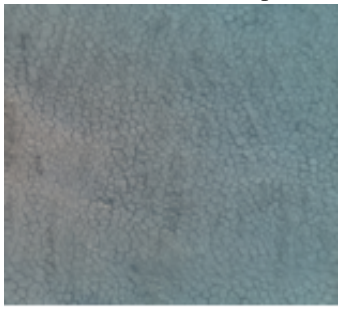


Apply of algorithm DexiNed after the oil spill

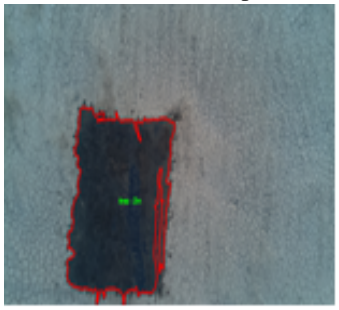


Area in pixel for all detected contours before the oil spill

Area in pixel for all detected contours after the oil spill



Applied the LAB algorithm before the oil spill and find the real area of the spill



Applied the LAB algorithm and find the real area of the spill after the oil spill

TABLE II.
STATISTICAL ANALYSIS OF A GROUP OF IMAGES CAPTURED IN PIXELS

Height=5 and The spill is ($1m^2$)		
image	**pixel area**	**perimeter area**
1	**101204.5**	**2478.83**
2	**97713.5**	**1816.19**
3	**96422.0**	**1568.59**
4	**87357.0**	**1384.76**
5	**91652.0**	**1373.94**
Height=10 and the spill is ($1m^2$)		
1	**23219.0**	**725.41**
2	**23019.5**	**797.25**
3	**23103.5**	**814.28**
4	**23099.0**	**762.38**
5	**23122.0**	**792.52**
Height=15 and the spill is ($1m^2$)		
1	**8078.0**	**457.1**
2	**7610.5**	**363.12**
3	**8760.0**	**499.16**
4	**8342.0**	**431.91**
5	**8041.0**	**393.42**
Height=5 and the spill is ($2m^2$)		
image	**pixel area**	**perimeter area**
1	**158437.0**	**2032.64**
2	**153535.0**	**2030.09**
3	**158332.5**	**2015.99**
4	**184341.0**	**2309.71**
5	**162207.5**	**2083.99**
Height=10 and the spill is ($2m^2$)		
1	**61926.0**	**1170.63**
2	**63696.0**	**1396.94**
3	**64244.5**	**1311.79**
4	**60996.5**	**1245.02**
5	**59781.5**	**1230.73**
Height=15 and the spill is ($2m^2$)		
1	**27382.5**	**767.71**
2	**27901.0**	**753.47**
3	**27481.0**	**711.04**
4	**27127.0**	**703.39**
5	**25840.0**	**864.1**
Height=5 and the spill is ($3m^2$)		
image	**pixel area**	**perimeter area**
1	**259915.0**	**2758.26**
2	**254728.0**	**2932.97**
3	**254765.0**	**3213.02**
4	**251951.0**	**3627.29**
5	**261107.5**	**3273.14**
Height=10 and the spill is($3m^2$)		
1	**143821.0**	**2523.32**
2	**138306.5**	**1932.6**
3	**135590.5**	**2400.83**
4	**138213.0**	**2475.24**
5	**138131.5**	**2364.26**
Height=15 and the spill is ($3m^2$)		
image	**pixel area**	**perimeter area**
1	**68020.5**	**1916.86**
2	**62248.0**	**1232.75**
3	**61547.5**	**1240.65**
4	**62929.5**	**1296.65**
5	**63246.5**	**1229.6**

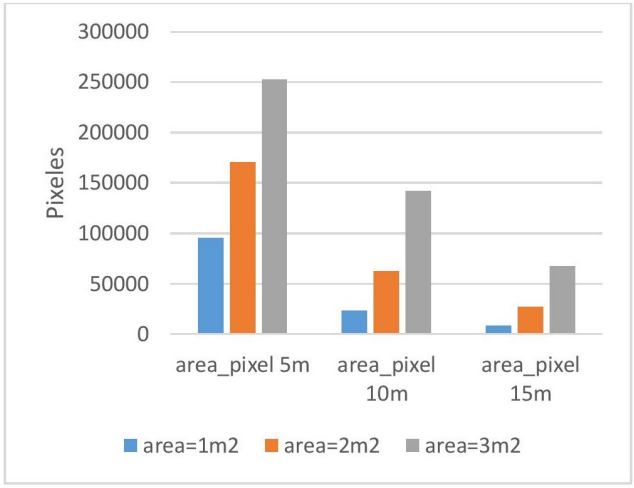


Fig. 9. Average pixel area of a set of leakage images.

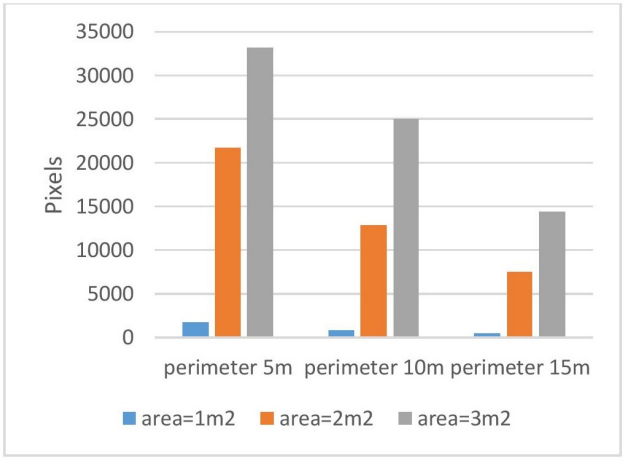


Fig. 10. Average perimeter of a set of leakage images.

VII. CONCLUSIONS

This work proposed designing and testing Iraq's crude oil pipelining system's monitoring system using a drone image processing and deep learning technique. The proposed system can enrich this type of research through a fully automated system. Extensive testing experiments were applied for different emulated spills in different locations. The proposed system is

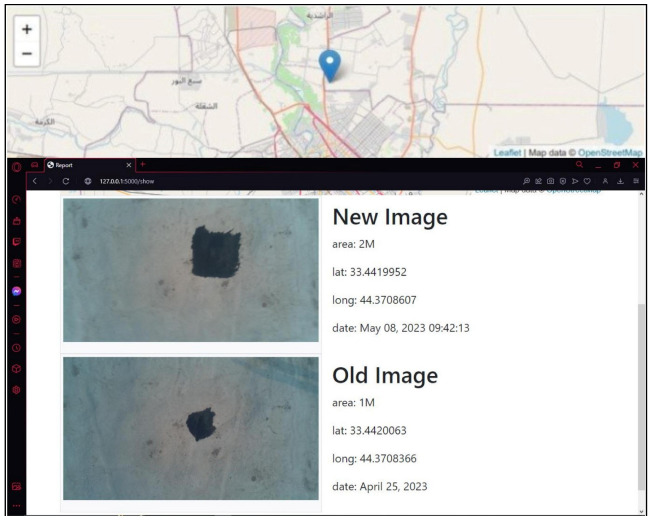


Fig. 11. Main program interface for oil spill expansion.

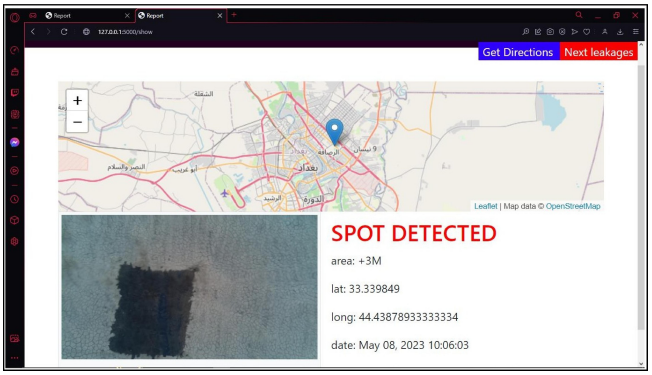


Fig. 12. Main interface of the program for the presence of an oil spill is greater than the threshold limit.

more accurate than traditional methods used in this field, such as differential pressure, where it cannot provide the position of the spill and provide only a leakage or not. The main difficulty of the proposed system is the period when the drone can fly to the monitor's longest path since some pipe segments reach 600km in Iraq. This case was considered a future work such that a drone swarm could be used to accomplish the work.

Conflict of Interest

The authors have no conflict of relevant interest to this article

Algorithm 1: Calculate the area of the oil leakage

```
Start a round of drones
{
Read the image and GPS from the drone.
Sending images and GPS via the MQTT IoT protocol through the cloud to the base station.
Find the edge of the contour in the image using DexiNed.
L * A * B * algorithm to select contour with black color only
Compute the area and perimeter of the suspicious contour
If (Perimeter >Perimeter_ threshold)
{
Send alarm message
}
Else
{
if (Perimeter <Perimeter_ threshold AND Perimeter >0) {
Save the image in a database for the second road of detection
}
}
END
```

References

[1] S. Adenubi, D. Appah, E. Okafor, and V. Aimikhe, "A review of leak detection systems for natural gas pipelines and facilities," 2023.

[2] K. G. Aljuaid, M. A. Albuoderman, E. A. AlAhmadi, and J. Iqbal, "Comparative review of pipelines monitoring and leakage detection techniques," in *2020 2nd International Conference on Computer and Information Sciences (ICCIS)*, pp. 1–6, IEEE, 2020.

[3] M. A. Adegboye, W.-K. Fung, and A. Karnik, "Recent advances in pipeline monitoring and oil leakage detection technologies: Principles and approaches," *Sensors*, vol. 19, no. 11, p. 2548, 2019.

[4] L. Boaz, S. Kaijage, and R. Sinde, "An overview of pipeline leak detection and location systems," in *Proceedings of the 2nd Pan African International Conference on Science, Computing and Telecommunications (PACT 2014)*, pp. 133–137, IEEE, 2014.

[5] S. Marathe, "Leveraging drone based imaging technology for pipeline and rou monitoring survey," in *SPE Asia Pacific Health, Safety, Security, Environment and Social Responsibility Symposium?*, p. D021S006R001, SPE, 2019.

[6] J. Allen and B. Walsh, "Enhanced oil spill surveillance, detection and monitoring through the applied technology

of unmanned air systems," in *International oil spill conference*, vol. 2008, pp. 113–120, American Petroleum Institute, 2008.

[7] T. Bakirman, B. Kulavuz, and B. Bayram, "Use of artificial intelligence toward climate-neutral cultural heritage," *Photogrammetric Engineering & Remote Sensing*, vol. 89, no. 3, pp. 163–171, 2023.

[8] M. Fahimipirehgalin, E. Trunzer, M. Odenweller, and B. Vogel-Heuser, "Automatic visual leakage detection and localization from pipelines in chemical process plants using machine vision techniques," *Engineering*, vol. 7, no. 6, pp. 758–776, 2021.

[9] O. Ibitoye, O. Shafiq, and A. Matrawy, "A convolutional neural network based solution for pipeline leak detection," *Recruit Researchers, Oct*, 2019.

[10] A. Li, D. Ye, E. Lyu, S. Song, M. Q.-H. Meng, and C. W. de Silva, "Rgb-thermal fusion network for leakage detection of crude oil transmission pipes," in *2019 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 883–888, IEEE, 2019.

[11] Z. Jiao, G. Jia, and Y. Cai, "A new approach to oil spill detection that combines deep learning with unmanned aerial vehicles," *Computers & Industrial Engineering*, vol. 135, pp. 1300–1311, 2019.

[12] M. Krestenitis, G. Orfanidis, K. Ioannidis, K. Avgerinakis, S. Vrochidis, and I. Kompatsiaris, "Oil spill identification from satellite images using deep neural networks," *Remote Sensing*, vol. 11, no. 15, p. 1762, 2019.

[13] A. Alharam, E. Almansoori, W. Elmadeny, and H. Alnoiami, "Real time ai-based pipeline inspection using drone for oil and gas industries in bahrain," in *2020 International Conference on Innovation and Intelligence for Informatics, Computing and Technologies (3ICT)*, pp. 1–5, IEEE, 2020.

[14] P. Ravishankar, S. Hwang, J. Zhang, I. X. Khalilullah, and B. Eren-Tokgoz, "Darts—drone and artificial intelligence reconsolidated technological solution for increasing the oil and gas pipeline resilience," *International Journal of Disaster Risk Science*, vol. 13, no. 5, pp. 810–821, 2022.

[15] A. A. Sharafutdinov, F. S. Khafizov, I. F. Khafizov, A. V. Krasnov, A. V. Akhmethafizov, V. I. Zakirova, and A. N. Khafizova, "Development of a method for calculating fire and oil spills parameters," in *AIP Conference Proceedings*, vol. 2216, AIP Publishing, 2020.

[16] M. Mahdianpari, B. Salehi, F. Mohammadimanesh, G. Larsen, and D. R. Peddle, "Mapping land-based oil spills using high spatial resolution unmanned aerial vehicle imagery and electromagnetic induction survey data," *Journal of applied remote sensing*, vol. 12, no. 3, pp. 036015–036015, 2018.

[17] N. V. S. Korlapati, F. Khan, Q. Noor, S. Mirza, and S. Vaddiraju, "Review and analysis of pipeline leak detection methods," *Journal of Pipeline Science and Engineering*, p. 100074, 2022.

[18] N. V. S. Korlapati, F. Khan, Q. Noor, S. Mirza, and S. Vaddiraju, "Review and analysis of pipeline leak detection methods," *Journal of Pipeline Science and Engineering*, p. 100074, 2022.

[19] Y. Zhou, H. Zhao, and Y. Liu, "An evaluative review of the vtol technologies for unmanned and manned aerial vehicles," *Computer Communications*, vol. 149, pp. 356–369, 2020.

[20] S.-M. Hou, C.-L. Jia, Y.-B. Wanga, and M. Brown, "A review of the edge detection technology," *Sparklinglight Transactions on Artificial Intelligence and Quantum Computing (STAIQC)*, vol. 1, no. 2, pp. 26–37, 2021.

[21] T. Bakirman, B. Kulavuz, and B. Bayram, "Use of artificial intelligence toward climate-neutral cultural heritage," *Photogrammetric Engineering & Remote Sensing*, vol. 89, no. 3, pp. 163–171, 2023.

[22] X. Soria, A. Sappa, P. Humanante, and A. Akbarinia, "Dense extreme inception network for edge detection," *Pattern Recognition*, vol. 139, p. 109461, 2023.

[23] R. Grompone von Gioi and G. Randall, "A brief analysis of the dense extreme inception network for edge detection," *IPOL. Journal Image Processing On Line, no 12, Oct 2022, pp. 389-403.*, 2022.

[24] X. Soria, A. Sappa, P. Humanante, and A. Akbarinia, "Dense extreme inception network for edge detection," *Pattern Recognition*, vol. 139, p. 109461, 2023.

[25] Z. Tian, D. Chen, S. Liu, and F. Liu, "Dexined-based aluminum alloy grain boundary detection algorithm," in *2020 Chinese Control And Decision Conference (CCDC)*, pp. 5647–5652, IEEE, 2020.

[26] Z. Tian, D. Chen, S. Liu, and F. Liu, "Dexined-based aluminum alloy grain boundary detection algorithm," in *2020 Chinese Control And Decision Conference (CCDC)*, pp. 5647–5652, IEEE, 2020.

[27] L. Alzubaidi, J. Zhang, A. J. Humaidi, A. Al-Dujaili, Y. Duan, O. Al-Shamma, J. Santamaría, M. A. Fadhel, M. Al-Amidie, and L. Farhan, "Review of deep learning: Concepts, cnn architectures, challenges, applications, future directions," *Journal of big Data*, vol. 8, pp. 1–74, 2021.

[28] R. Grompone von Gioi and G. Randall, "A brief analysis of the holistically-nested edge detector," *IPOL. Journal Image Processing On Line, no 12, Oct 2022, pp. 369-377*, 2022.

[29] N. Yu, Z. Zhang, Q. Xu, E. Firdaous, and J. Lin, "An improved method for cloth pattern cutting based on holistically-nested edge detection," in *2021 IEEE 10th Data Driven Control and Learning Systems Conference (DDCLS)*, pp. 1246–1251, IEEE, 2021.

[30] I. Misra, M. K. Rohil, S. M. Moorthi, and D. Dhar, "Sprint: Spectra preserving radiance image fusion technique using holistic deep edge spatial attention and minnaert guided bayesian probabilistic model," *Signal Processing: Image Communication*, vol. 113, p. 116920, 2023.

[31] L. Dong, W. Zhang, and W. Xu, "Underwater image enhancement via integrated rgb and lab color models," *Signal Processing: Image Communication*, vol. 104, p. 116684, 2022.

[32] N. Abbadi and E. Razaq, "Automatic gray images colorization based on lab color space," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 18, no. 3, pp. 1501–1509, 2020.

[33] B. Mishra and A. Kertesz, "The use of mqtt in m2m and iot systems: A survey," *IEEE Access*, vol. 8, pp. 201071–201086, 2020.

[34] S. S. Ibraheem, A. H. Hamad, and A. S. A. Jalal, "A secure messaging for internet of things protocol based rsa and dna computing for video surveillance system," in *2018 Third Scientific Conference of Electrical Engineering (SCEE)*, pp. 280–284, IEEE, 2018.