

## Application of CNN based image classification technique for oil spill detection

K Das<sup>\*a</sup>, P Janardhan<sup>a</sup> & H Narayana<sup>b</sup>

<sup>a</sup>Department of Civil Engineering, National Institute of Technology Silchar, Assam – 788 010, India

<sup>b</sup>Department of Civil Engineering, National Institute of Technology Goa, Farmagudi, Goa – 403 401, India

\*[E-mail: koushik\_rs@civil.nits.ac.in]

*Received 22 September 2022; revised 19 January 2023*

Marine water pollution due to oil spills is a common threat to the environment worldwide because of its harmful impact on the economy and environment. Remote Sensing (RS) and Geographic Information Systems (GIS) are well-known tools for collecting satellite data which helps in remote oil spill identification. Synthetic Aperture Radar (SAR) images through various satellite missions are the mainly used data to identify oil spills. Many Artificial Neural Networks (ANN) and Machine Learning (ML) models integrated with RS and GIS have been originated and applied to identify and monitor oil spills. Deep Learning (DL) methods have recently become popular for their outstanding performance in research for image classification challenges, and the same is being used in the present study. An oil spill detection model using the Convolutional Neural Network (CNN) algorithm is presented in this work. CNN can extract features from a large dataset, and these features can be used to categorize images into different classes. The proposed model was compared with other existing models. The accuracy, precision, and recall achieved by this study are 99.06 %, 98.15 %, and 100 %, respectively. The proposed model outperformed the other existing work with an accuracy of 99.06 % and a precision of 98.15 %.

**[Keywords:** Convolutional Neural Network, Geographic Information System, Image classification, Oil spill, Synthetic Aperture Radar]

### Introduction

An oil spill is one type of pollution caused by human activities that harm marine life. It can be a disastrous consequence for society, both economically and environmentally. In the short- or long-term, oil spills create dangerous results on the earth, like the death of flora and fauna, especially the deaths of marine mammals such as whales, dolphins, seals, etc. It also changes the life cycles of plants and animals, the coastal ecosystem, etc.<sup>1</sup>. The main reasons behind oil spills are the release of unrefined petroleum products during the oil production process and ship accidents during the transportation of oils through shipping<sup>2-5</sup>.

The location of the catastrophes that caused oil spills in the past indicates that most of the accidents occurred in areas of complex oceanic environments; and because of this, it is not so easy to go to the areas directly to clean the oil and observe the spread of oil during the early stages of the incidents. As manual detection of oil spills is not feasible, it is vital to use satellite images for detecting and observing the spread of oil.

An active sensor like Synthetic Aperture Radar (SAR) can cover a large area of oil spills irrespective of time and weather, which has advantages in oil spill

detection compared to spaceborne optical and infrared sensors<sup>6</sup>. Remote sensors mainly used for detecting oil spills are laser fluorosensors, microwave sensors, and optical remote sensors. Microwave sensors are prevalent for detecting oil spills, specifically active sensors such as radars<sup>7</sup>. The main satellites used to monitor oil spills are SEASAT, ALMAZ, ERS-1 & 2, RADARSAT-1 & 2, ENVISAT, TerraSAR-X, COSMO-SkyMed, Cloudsat, RISAT-2, TanDEM-X, RISAT-1, Metop-B, Sentinel-1 and 2, etc.<sup>4,8-11</sup>. Also, Side-Looking Airborne Radar (SLAR) images can be used for oil spill detection<sup>12</sup>.

The dark regions present in SAR imagery indicate the oil-covered areas of the sea surface due to the scattering of the SAR signal from the oiled surface. The dark regions or spots can be oil spills or look-alikes<sup>13-15</sup>. The main three steps of oil spill detection are: (a) identifying the dark spots, (b) extracting features from these dark spots, and (c) training a classifier by using the extracted features<sup>16</sup>. Then this obtained classifier can be applied to classify or categorize the selected dark regions<sup>2</sup>.

For dark spot detection, various methods like thresholding, edge detection, multilayer perceptron,

image pyramid, etc., have been used in different studies<sup>16-22</sup>. After dark spot detection, the spectral, geometrical, contextual, and other characteristics of the dark spots has been used as an input to the classifier<sup>21,23</sup>. Many image classification algorithms based on traditional ML have been introduced for dark spot identification and classification using SAR images in the last two decades. Frate *et al.*<sup>17</sup> presented a study where multilayer perceptron was used as network topology, whereas Fiscella *et al.*<sup>18</sup> used different classifiers. Many more methods have been applied in different studies related to oil spills<sup>20,24-27</sup>.

In the last few years, some classification methods based on DL algorithms have been used for detecting oil spills. A CNN was used by Guo *et al.*<sup>28</sup> and obtained an accuracy of 91.3 %. Hidalgo *et al.*<sup>12</sup> has used a two-stage CNN and obtained accuracy, precision and recall of 98 %, 52 % and 73 %, respectively. Gallego *et al.*<sup>29</sup> applied Residual Encoder-Decoder Network (RED-Net) for oil spill segmentation using Side-Looking Airborne Radar (SLAR) images. A DL system can easily extract features in an automated way from the dataset, whereas traditional feature learning models involve hand-coded features.

SAR images with oil spill areas are highly preferred for oil spill mapping on the Sentinel Application Platform (SNAP). The dark spots in SAR imagery are considered oil spills because oil fields reduce the unevenness of the sea surface, which enhances the backscattering from the surface and causes spills to look darker in SAR imagery compared to spill-free areas. But all dark patches in a SAR image are not always oil spills; look-alikes also appear as dark spots. So, it is necessary to collect SAR images containing oil spills that can be used to map them. In this study, the discrimination of SAR

images in two classes: an oil spill and without an oil spill, has been done. For binary and multiclass image classification, CNN is a widely used algorithm, and hence the SAR images have been classified using CNN for this study. The data used for this study was collected from Krestenitis *et al.*<sup>30</sup>. The provided dataset by Krestenitis *et al.*<sup>30</sup> contains pre-processed SAR imagery with an annotated image of the same. The SAR images are then split into training and testing datasets based on the corresponding annotated images to train and test the proposed CNN.

## Materials and Methods

### Dataset

The dataset collected from Krestenitis *et al.*<sup>30</sup> contains 1112 images, including oil spill and non-oil spill images and corresponding annotated images. A SAR image and its corresponding annotated image are shown in Figure 1. In their earlier study, Krestenitis *et al.*<sup>31</sup> has annotated the images manually. In the annotated image in Figure 1b, oil spills are represented by cyan, look-alikes in red, ships in brown, land in green, and sea surface in black. The European Maritime Safety Agency (EMSA) records and human identification helped create the annotated image sets to semantically mark the different regions of the images with a specific colour. The data was acquired by Krestenitis *et al.*<sup>30</sup> from the Copernicus Open Access Hub, a European Space Agency database. The Sentinel-1 satellite mission collected these datasets between September 28 and October 31, 2015.

The GMES system of the European Union has conducted this Sentinel-1 SAR mission to collect highly featured remote sensing data to monitor various crucial problems related to marine and coastal environments. The satellites for this mission are

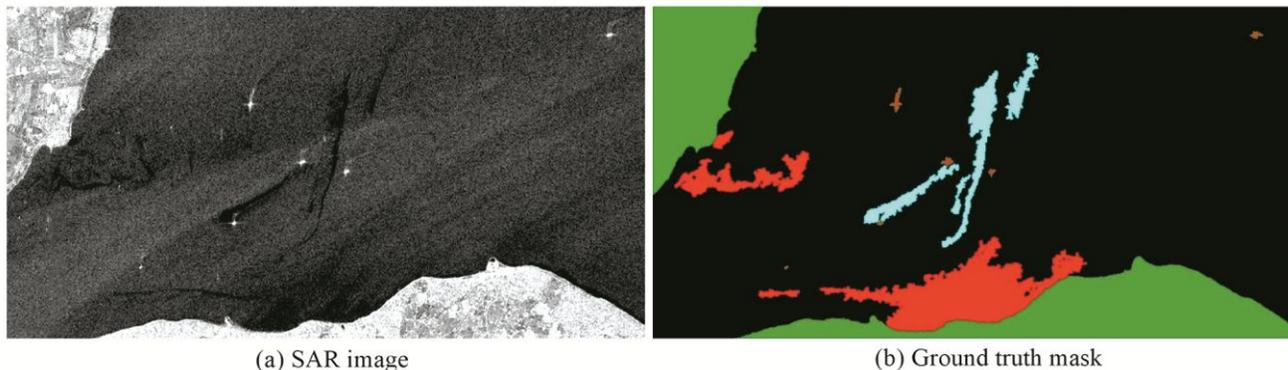


Fig. 1 — (a) SAR image, and (b) corresponding annotated image where oil spills are represented by cyan, look-alikes by red, ships by brown, land by green, and sea surface by black

outfitted with the SAR system, which operates in horizontal and vertical polarisations at C-band.

The satellite images obtained by this mission have the following polarisation: vertical polarisation transmitted—vertical polarisation received (VV) and vertical polarisation transmitted—horizontal polarisation received (VH). Krestenitis *et al.*<sup>30</sup> have processed the SAR data of the VV band for creating the SAR image dataset. The obtained dataset was already pre-processed by Krestenitis *et al.*<sup>30</sup>. The pre-processing steps are:

- a) Tracing of all reported oil spills using the EMSA records.
- b) Subsets of the raw images have been generated to reduce the processing time and file size of the processed datasets. These images have been rescaled to get a resolution of 1250×650 pixels.
- c) Radiometric calibration has been applied to all of these images.
- d) Then speckle filtering was done for these images to improve the contrast and reduce noise.
- e) Then, a transformation function converted dB to actual radiance values.

The ground truth masks have been used for dividing the SAR images of training and test sets into with and without an oil spill, and then the SAR images have been used for classification using the CNN algorithm.

#### Training and validation of data

A total of 466 images have been used for this study, of which 360 were used as training data and 106 as test data. The algorithm was trained and adjusted using the training dataset. The initial size of the images of both the training and test datasets was 1250×650 pixels. The input size of the images used for the algorithm was reduced to 224×224 pixels, reducing the computational load of the method and making the process easier<sup>30,32,33</sup>. The training images were the input to the CNN algorithm, where the algorithm extracted the important features from these images to create a vector matrix of features. This feature matrix is used as an input to the classification layers of CNN. After training and adjusting the algorithm, the model's performance was tested using the test dataset.

#### The basic structure of CNN

CNN is ideal for image-based object recognition and always gives the best results for image classification. CNN transforms the input data into a set of class scores through all its layers, from the input

to the output. The basic architecture of CNN is based on the following three layers - an input layer, feature-extraction layers, and classification layers. CNN can learn useful representations for a job on its own. Every convolutional layer can learn and respond to a specific feature. Deeper layers use combinations learned from previous layer outputs to represent more abstract aspects<sup>34</sup>.

Input layers are where the raw input images are stored for processing in the network. The convolutional layer is the most important component in CNN architecture, where the features of the raw input image are extracted, and it gives an output feature map. The convolutional layer consists of the following components - Kernel matrix, strides and padding. The Kernel is the filter used to extract the features from input images by moving across the images and performing dot products between the Kernel filter and the sub-region of the input image. The stride value represents the amount of pixel shifting over the input image, which is always almost symmetrical in height and width dimensions. Every time a convolution process is conducted, the output image shrinks.

Furthermore, pixels in the corners and edges are used less frequently than those in the middle. Padding is usually done in CNN to solve this problem. The equation to express the size of a feature map is:

$$M = (N + 2p - f)/S + 1 \quad \dots (1)$$

Where,  $M$ ,  $N$ , and  $f$  are the feature map size, input image size, and size of convolution kernel, respectively;  $S$  is the stride, and  $p$  is the padding value.

Generally, the layer added next to the convolutional layer is the pooling layer. The function of a pooling layer is to down sample the output feature map<sup>35</sup>. The reduction of the size of the feature map means gathering more meaningful data by eliminating the noise of the data and extracting essential data. The mostly used pooling method is max pooling among the available three methods: max pooling, average pooling, and sum pooling. Although before adding the pooling layer, it is necessary to add non-linearity to the network for extracting features, which is the role of activation functions in CNN. These functions govern the behaviour of artificial neurons. Activation functions decide whether a neuron should be activated or not. The most commonly used activation functions are Rectified Linear Unit (ReLU), leaky ReLU, sigmoid, softmax, etc.<sup>34</sup>.

Generally, the fully connected layer is the last layer of a CNN architecture. Its structure is the same as the fully connected network of traditional neural network models. After getting the pooled feature map from the pooling layer or the last convolutional layer, the next step is flattening the feature map for further processing. For this, a reshaped layer or flattened layer is used in CNN. This flattened layer converts the feature map into a 1-dimensional vector, which is the input of the fully connected layer of CNN. The output of this fully connected layer represents the final CNN output<sup>35</sup>.

### Proposed methodology

#### *Proposed oil spill detection algorithm*

CNN algorithm has been used in this study for classifying the SAR images. The objective of a CNN algorithm is to learn the important features of the dataset automatically and use these features for classifying or predicting their class.

This study used four convolutional layers to train the model, and after every convolutional layer, a pooling layer was used. Altogether, sixty-four filters were used in the first layer, with the kernel size being 3×3. For this layer, the input shape of the image was taken as 224×224×3. After passing through the first convolution layer, the output shape of the layer has become 224×224×64. Once the first convolutional layer creates feature maps, these are converted into pooled feature maps of size 112×112×64 while passing through a max-pooling layer of a 2×2 matrix. The width of the convoluted and pooled feature maps is the same because the same or zero padding is applied in this study. The ReLU activation function has been used in the convolutional layers in the present study.

In the subsequent two convolutional layers, 64 filters have been used, whereas, in the last layer, 128 filters are used. The number of pooled feature maps after every layer is the same as the number of convolution filters used. However, the size is reduced

by half as a max-pooling layer of a 2×2 matrix has been used after the convolution layers. As 128 filters have been used at the last convolutional layer, the size of pooled feature map achieved by this model is 14×14×128. The steps involved in the feature extraction layers are repeated to get a small feature map as the final shape. The kernel matrix used for all these layers has a shape of 3×3. All the convolved matrices achieved from all these convolutional layers have passed through a max-pooling layer of 2×2 size.

This study aims to develop a classification model that classifies the images into two categories by taking the feature map as input in a 1-dimensional vector for classification. To convert the outputs, *i.e.*, pooled feature maps to a 1-dimensional linear vector, reshaping has been done using the flattened layer. In this layer, the pooled feature map has been flattened, and the output of this layer is used for the fully connected layer, which has 128 filters. The ReLU activation function has been used for the hidden layer of this fully connected layer, and a dropout of 0.5 has been applied. A node is activated by the ReLU activation function only when the input value is above a threshold limit. It converts negative values from the convolved matrix to zero values and adds non-linearity to CNN.

The fully connected layer has the same structure as an artificial neural network, where the flattened vector of the feature map undergoes a few more layers, and the classification occurs in the output layer. The sigmoid has been used as the activation function in this output layer. For a classification problem, the output is the probability of a specific class that the model predicts. As the probability of anything exists only between 0 and 1, a sigmoid is correct for this problem. The detailed structure of the proposed network is shown in Figure 2. The filter size, filter number, and stride value used for the layers used in this CNN are shown in Table 1.

Table 1 — Structural details of the proposed network

Layer	Layer type	Filter size	Filter number	Stride
Conv1	Convolution	3	64	1
Pooling1	Maxpooling	2	-	2
Conv2	Convolution	3	64	1
Pooling2	Maxpooling	2	-	2
Conv3	Convolution	3	64	1
Pooling3	Maxpooling	2	-	2
Conv4	Convolution	3	128	1
Pooling3	Maxpooling	2	-	2
Flatten	Flatten	-	-	-
Fully connected	Fully connected	128	-	-
Sigmoid	Sigmoid	1	-	-

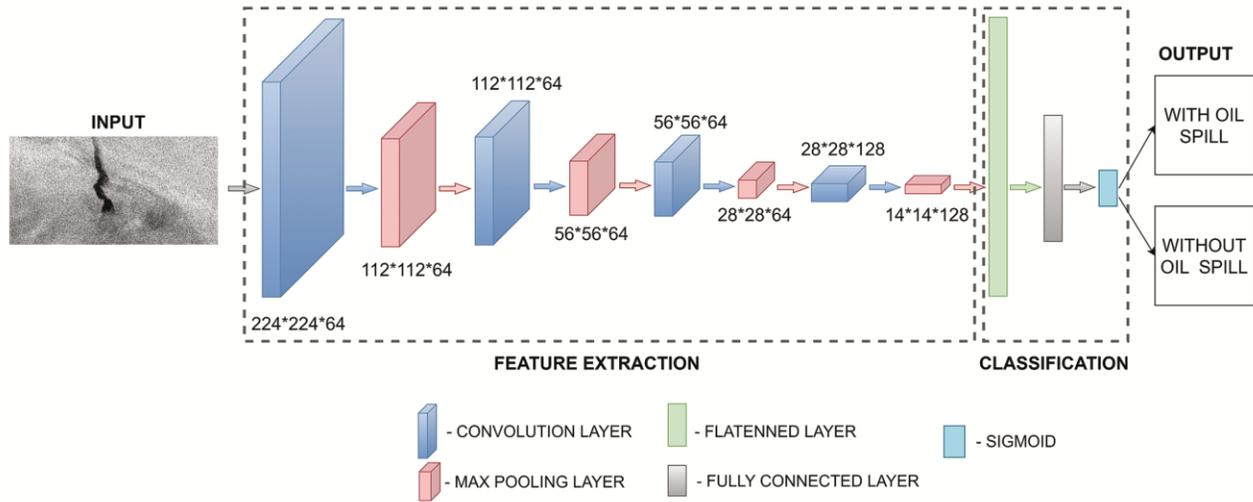


Fig. 2 — Architecture of the proposed CNN

If a model is trained with a lot of data, it starts learning from the noise and defective data present in the dataset. The model extracts features from these defective datasets, and the number of adjustable parameters increases from the dataset; because of that, the model's performance decreases for classifying the new dataset. This issue is caused by an over adjustment to specific training data, which includes random character variances that cannot be extrapolated to new data<sup>32</sup>. This problem is called the overfitting of the model. Many regularization techniques are used to enhance the adapting ability of the model to control unknown data<sup>36</sup>. Dropout is one of these techniques, which removes the neurons randomly during the model's training to avoid additional co-adaptation in neurons<sup>37</sup>. To reduce overfitting, a dropout of 0.25 has been used in the last three layers of feature extraction layers, but for flattening, a dropout of 0.5 has been used in this study.

Dataset augmentation is a method to increase the generalization capacity of a network for better performance and accuracy of the model by applying some transformation techniques to the dataset. It increases the quantity of the dataset by forming a changed and modified version of the dataset without influencing their classes<sup>36</sup>. Data augmentation is a broadly used technique for building a deep learning model in many studies<sup>32,33</sup>. For this study, different data augmentation techniques have been carried out over the training set, such as rotation of images using a rotation range of 10, horizontal flip, a zoom range transformation using a zoom range value of 0.2, shear

range transformation using a shear range value of ten and rescaling, but for the test set only rescaling has been done.

A loss function quantifies the error of prediction in a classification algorithm. It evaluates the error between the output and the target value. As this study estimates the probability for binary classification, binary cross-entropy has been used as a loss function<sup>34</sup>.

Optimizers are the algorithms used to reduce the error or loss function or increase the model's efficiency. The learnable parameters of the model, *i.e.*, weights and biases, are the factors that control these mathematical functions. The Adam optimizer is used for this study, giving better results than any other optimization algorithm. It also takes less computation time and fewer parameters for tuning.

The structure of the network is determined by the hyperparameters, which also determine the learning rate of the network. These parameters are tuned to make the networks train better and faster. The hyperparameter selection ensures that the model does not underfit or overfit the training dataset while learning the data structure.

Batch size and epoch are two important hyperparameters in deep learning algorithms. Batch size represents the number of samples from the training dataset that are processed during training the model. At the same time, an epoch is the total number of passes required for a complete cycle of the training dataset. Epochs are comprised of batch size and iterations. The model does not develop the value of these parameters as they are not the algorithm's

internal parameters, so they are specified during the training process. These values can be changed during algorithm training to obtain the best model. Although a single epoch completes the whole training dataset, it is not enough to obtain a fit model and causes the underfitting of the model. The model was trained for different epochs, such as 30, 50, and 75, to observe the difference in performance. The difference between the model's performance for 30 and 50 epochs is discussed in the results and discussion part and is shown in Figure 3.

#### Evaluation of model

The correctness of classification and prediction in a particular context can be understood by evaluating a model. The important measures or metrics for evaluating the model performance are determined from the confusion matrix of a model. A confusion matrix evaluates the performance of a model by representing the predicted and ground truth class in the matrix. This table is used for a better understanding of the performance of a model or a classifier from the following terms:

True Positives (TP): when the model correctly classified a label.

False Positives (FP): when the model predicted a negative label as a positive label.

True Negatives (TN): when the negative labels are

predicted as negative.

False Negatives (FN): when the positive label is predicted as negative.

## Results and Discussion

### Experimental results and analysis

The accuracy and loss of the training and test datasets are shown in Figures 3 to 4. Figure 3(a) represents the model accuracy for the training and test datasets when the number of epochs used in the model are 30. The figure shows that the accuracy increases rapidly in the first three epochs for the training set, indicating fast learning of the network. Afterwards, it increases gradually until the 8<sup>th</sup> epoch, and then there is no such rise and fall in the accuracy curve for the training set. It also shows that the accuracy of the test dataset increases rapidly in the first three epochs, and then shows some undulations. This model achieves the highest training and test accuracy of 100 % and the lowest training and test accuracy of 57.22 % and 78.30 %, respectively, for 30 epochs. The figure shows that the training and test accuracy curves are almost identical, *i.e.*, the model performs well for both datasets. Figure 3(b) represents the loss of the model when the number of epochs is 30. It demonstrates that the training set's loss decreases rapidly in the first six epochs, then decreases

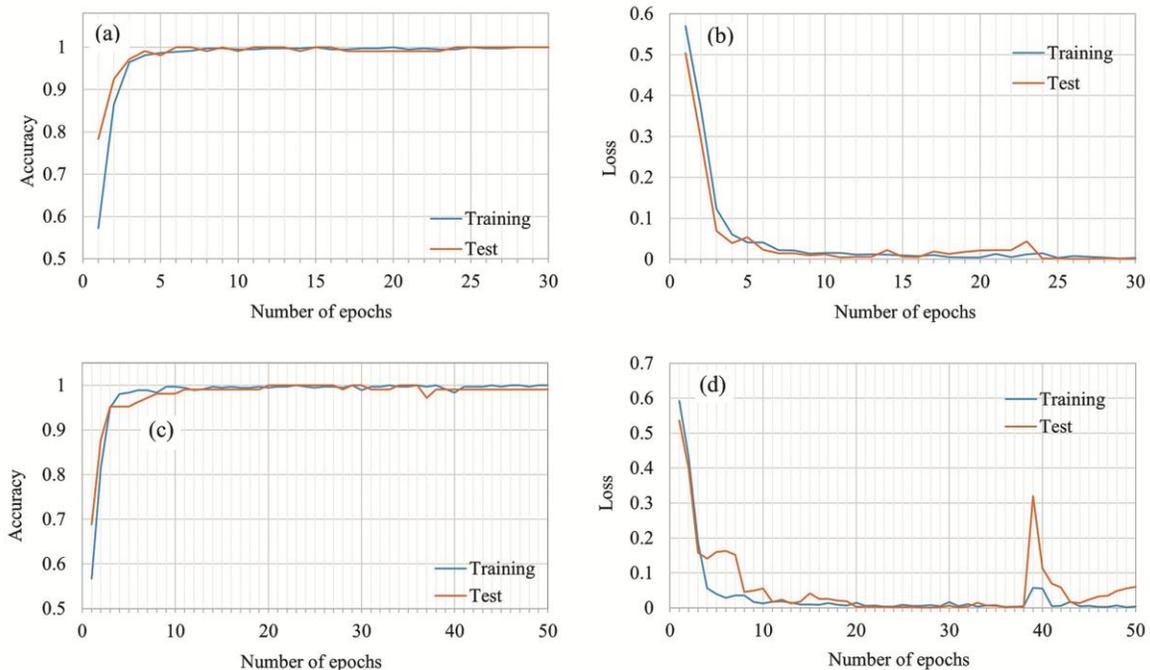


Fig. 3 — (a), (c) Model accuracy for 30 and 50 epochs, respectively; and (b), (d) model loss for 30 and 50 epochs, respectively

gradually till the 8<sup>th</sup> epoch and then there are no such differences in the curve. The loss decreases rapidly in the first four epochs for the test set, and then some undulations can be observed. It is evident from Figure 3 that there is no such difference in the model performance for 30 and 50 epochs, so it can be said that 30 epochs are better for this model as it takes less time compared to 50 epochs.

As mentioned in the proposed methodology that the dropouts have been used to reduce the overfitting of the model. Still, the model was trained for 30 and 50 epochs to observe the model’s performance without using dropout. Figure 4 represents the model accuracy and loss for 30 and 50 epochs without dropout. It can be seen from these figures that the test accuracy is higher than the training accuracy for both 30 and 50 epochs. These figures also show a sudden rise and fall in the accuracy and loss curves due to the absence of dropouts. Hence, a dropout is essential to avoid such circumstances.

Figure 5 represents the confusion matrix the proposed algorithm achieves for test data. The performance of the proposed CNN algorithm was evaluated using a total of 106 test images. It is clear from the confusion matrix that the model classifies 53 as true positive and 52 as true negative, *i.e.*, it has correctly classified all the oil spill images, but only one image without oil spill is classified as with oil spill, which is a false positive. Table 2 shows the various measurements derived from the confusion

Measure	Derivations	Value
Sensitivity	$TP / (TP + FN)$	1.0000
Specificity	$TN / (FP + TN)$	0.9811
Precision	$TP / (TP + FP)$	0.9815
Negative predictive value	$TN / (TN + FN)$	1.0000
False-positive rate	$FP / (FP + TN)$	0.0189
False discovery rate	$FP / (FP + TP)$	0.0185
False-negative rate	$FN / (FN + TP)$	0.0000
Accuracy	$(TP + TN) / (TP + TN + FP + FN)$	0.9906
F1 score	$2TP / (2TP + FP + FN)$	0.9907
Jaccard index	$TP / (TP + FP + FN)$	0.9813

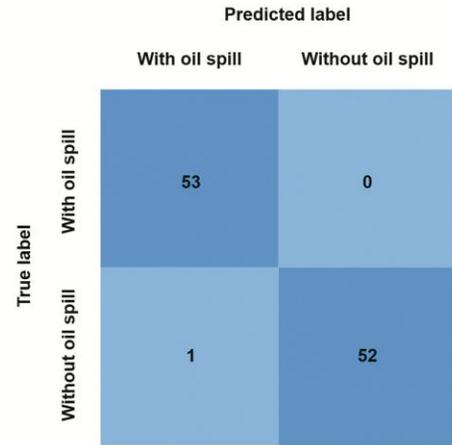


Fig. 5 — Confusion matrix of the CNN

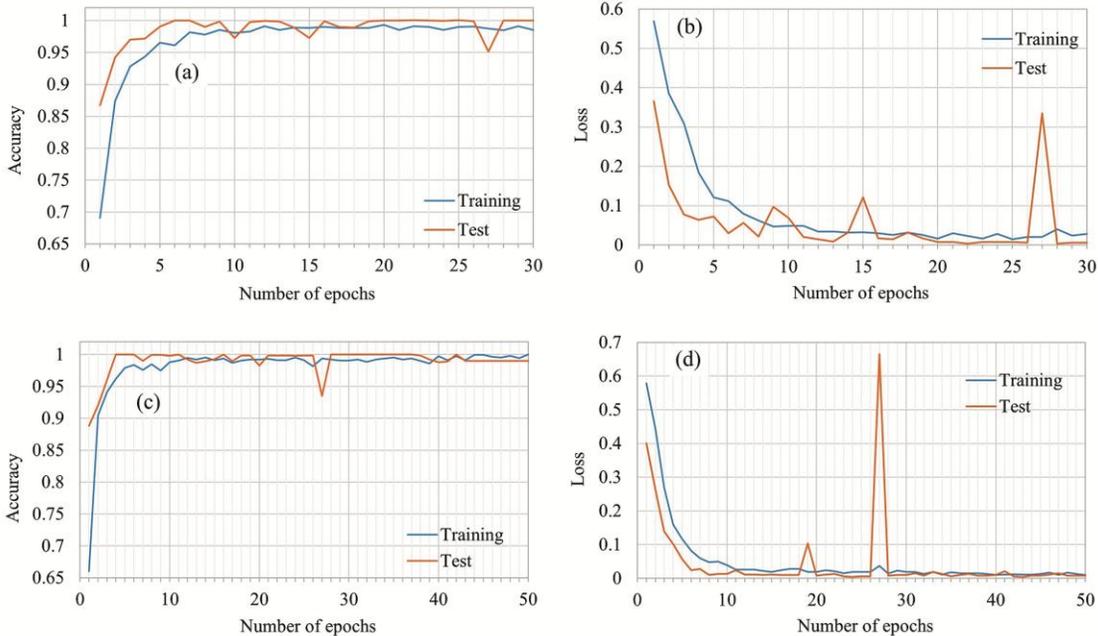


Fig. 4 — (a), (c) Model accuracy without dropout for 30 and 50 epochs, respectively; and (b), (d) Model loss without dropout for 30 and 50 epochs, respectively

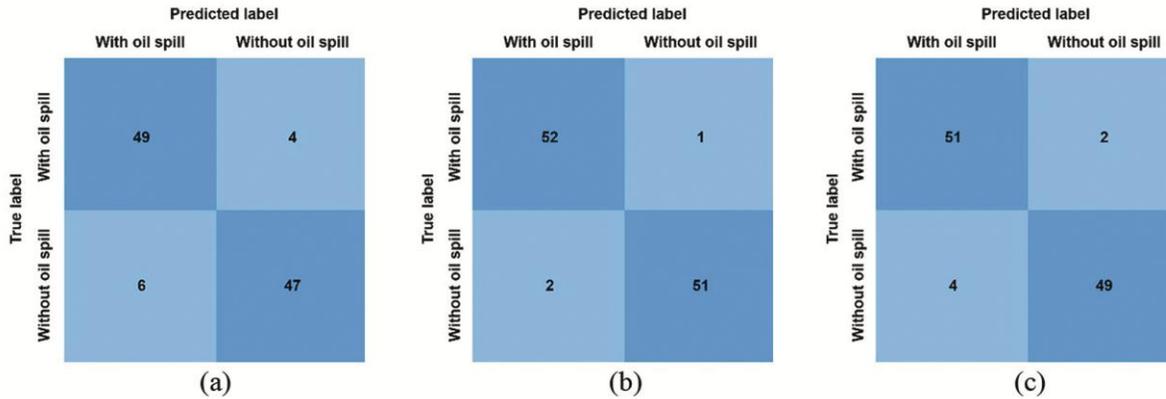


Fig. 6 — Confusion matrix achieved for the same dataset by using (a) AlexNet, (b) VGG16, and (c) VGG19

matrix to assess the model's performance. In this study, precision measures the correctness of the pixels that are correctly classified as oil among all the pixels classified as oil. This model's precision, accuracy, and F-1 score are 98.15, 99.06, and 99.07 %, respectively.

In this proposed model, the first convolution layer has 64 neurons, which extract 1792 parameters from the training dataset. Then by applying the max-pooling layer, the height of the convolutional layer was reduced, but the breadth remained the same. The second convolutional layer, which consists of 64 neurons, extracts 36928 parameters. The first convolution layer converts the input image into a featured map of size  $224 \times 224 \times 64$ , and then the featured map goes through a pooling layer which converts it into a pooled feature map of size  $112 \times 112 \times 64$ . The last pooled feature map passes through the last pooling layer and is converted into a size of  $14 \times 14 \times 128$ . Then the flattened layer converts the featured map into a 1-dimensional linear vector to use this vector as an input to the classification layer of the network. The network trains a total of 3,361,025 parameters by this network.

The performance of the proposed model was compared with three pre-trained networks: AlexNet, VGG16 and VGG19, by training and testing the same datasets for these networks. The input size of the images required to train the AlexNet is  $227 \times 227 \times 3$ , and for VGG16 and VGG19 is  $224 \times 224 \times 3$ . The images were resized before training the networks. These three networks gave the best result for 100 epochs (or 200 iterations). The confusion matrix achieved by these three models is shown in Figure 6, and the comparison of the performance metrics for the proposed model with these three pre-trained models is shown in Table 3.

Table 3 — Comparison of performance metrics of the proposed model with different networks

	Accuracy (%)	Precision (%)	Recall (%)
AlexNet	90.56	89.09	92.45
VGG16	97.16	96.29	98.11
VGG19	94.34	92.72	96.22
Proposed method	99.06	98.15	100

The Table 4 compares the proposed algorithm to existing oil spill classification models developed in previous studies.

This study has reduced the input images to  $224 \times 224$ , particularly for the following reasons: the proposed model was compared with three pre-trained models, and the input image size required for VGG 16 and VGG 19 is  $224 \times 224$ . So, for comparing the performance of the proposed model with these pre-trained models, the same input size of  $224 \times 224$  has been taken. Also, the feature learning process of the CNN-based models is independent of the aspect ratio of the input images. So, it is not mandatory to maintain the aspect ratio. The same features will be extracted from the images, as the features for the given task are independent of the aspect ratio of the images. Also, the computational time is reduced because of the reduced image sizes.

The proposed CNN model has outperformed the pre-trained models because the features for larger models are generalized on many different tasks. The features of the task at hand are comparatively simpler (shape, region and area). AlexNet, VGG 16 and VGG 19 models have pre-trained feature extraction layers unsuitable for the task as the models mainly work with colour images, but the oil spill data is mainly monochrome. So, in this scenario, a simple CNN model outperforms the larger trained architectures.

Table 4 — Comparison of the proposed work with existing work

	Method used	Activation function	Loss function	Optimization algorithm used	Accuracy (%)	Precision (%)	Recall (%)
The proposed model	CNN	ReLU* and sigmoid**	Binary cross entropy	Adam	99.06	98.15	100
Guo <i>et al.</i> , 2017	CNN	Sigmoid	-	-	91.33	-	-
Hidalgo <i>et al.</i> , 2018	CNN	ReLU* and softmax** (For four class CNN)	-	Stochastic gradient descent	96.64	61.15	60.97
		ReLU* and sigmoid** (For two-stage CNN)	-	Stochastic gradient descent	97.57	72.36	74.40
Cantorna <i>et al.</i> , 2019	CNN	ReLU* and sigmoid**	Binary cross entropy	RMSprop	98.3	85.70	92.7
Zeng <i>et al.</i> , 2020	CNN	ReLU	Softmax cross entropy	Adam	94.01	85.70	83.51
Song <i>et al.</i> , 2020	CNN + SVM	ReLU* and Softmax**	Cross entropy	Adadelta	99.19	95.87	98.12

\*For hidden layers, \*\* for output layers

## Conclusion

This paper proposes an enhanced deep CNN model for detecting oil spills using SAR image analysis. The data necessary to train the model was obtained from previous studies. The proposed methodology could accurately differentiate between oil spills and look-alikes with a quite high success rate in an automated way. ReLU and Sigmoid activation functions are used for hidden layers and output layers, respectively, in the present study. Thirty epochs with 0.5 dropouts were found sufficient in the proposed network to reduce the overfitting of the model. A confusion matrix is used to assess the model's performance. This model's precision, accuracy, and F-1 score are 98.15, 99.06, and 99.07 %, respectively. Proposed model outperformed the other existing work with an accuracy of 99.06 % and a precision of 98.15 %. The main advantage of the proposed work/model is that it can be used to manage similar environmental disasters, such as floods, soil erosion, deforestation, and so on, using relevant image samples.

## Acknowledgements

The authors would like to thank Marios Krestenitis for providing us with the data; and Director, National Institute of Technology Silchar, for the support and encouragement.

## Conflict of Interest

The authors declare that there is no conflict of interest.

## Ethical Statement

This material is the authors' own original work, which has not been previously published elsewhere.

## Author Contributions

KD modelled soft computing methods for classification and analysis and drafted the original manuscript. PJ & HN supervised and reviewed the manuscript.

## References

- Smith L C J, Smith M L & Ashcroft P A, Analysis of Environmental and Economic Damages from British Petroleum's Deepwater Horizon Oil Spill, *Albany Law Review*, 74 (1) (2011) 563-585. <https://dx.doi.org/10.2139/ssrn.1653078>
- Brekke C & Solberg A H S, Oil spill detection by satellite remote sensing, *Remote Sens Environ*, 95 (1) (2005) 1-13. <https://doi.org/10.1016/j.rse.2004.11.015>
- Solberg A H S, Brekke C & Husoy P O, Oil Spill Detection in Radarsat and Envisat SAR images, *IEEE Trans Geosci Remote Sens*, 45 (3) (2007) 746-754. <https://doi.org/10.1109/TGRS.2006.887019>
- Topouzelis K N, Oil Spill Detection by SAR Images: Dark Formation Detection, Feature Extraction and Classification Algorithms, *Sensors*, 8 (10) (2008) 6642-6659. <https://doi.org/10.3390/s8106642>
- Solberg A H S, Remote Sensing of Ocean Oil-Spill Pollution, *Proc IEEE*, 100 (10) (2012) 2931-2945. <https://doi.org/10.1109/JPROC.2012.2196250>
- Uhlmann S & Kiranyaz S, Classification of dualand single polarized SAR images by incorporating visual features, *ISPRS J Photogramm Remote Sens*, 90 (2014) 10-22. <http://dx.doi.org/10.1016/j.isprsjprs.2014.01.005>
- Fingas M & Brown C, Review of oil spill remote sensing, *Mar Pollut Bull*, 83 (1) (2014) 9-23. <https://doi.org/10.1016/j.marpolbul.2014.03.059>
- Kolokoussis P & Karathanassi V, Oil Spill Detection and Mapping Using Sentinel 2 Imagery, *J Mar Sci Eng*, 6 (1) (2018). <https://doi.org/10.3390/jmse6010004>
- Rajendran S, Vethamony P, Sadooni F N, Al-Kuwari H A S, Al-K J A, *et al.*, Detection of Wakashio oil spill off Mauritius using Sentinel-1 and 2 data: Capability of sensors, image transformation methods and mapping, *Environ Pollut*, 274 (2021) 116618. <https://doi.org/10.1016/j.envpol.2021.116618>

- 10 Mahmoud A S, Mohamed S A & El-K R A, Oil Spill Identification based on Dual Attention UNet Model Using Synthetic Aperture Radar Images, *J Indian Soc Remote Sens*, 51 (2023) 121–133. <https://doi.org/10.1007/s12524-022-01624-6>
- 11 Vijayakumar S, Computational Techniques of Oil Spill Detection in Synthetic Aperture Radar Data: Review Cases, *Recent Oil Spill Challenges That Require More Attention*, 5 (2023) <https://doi.org/10.5772/intechopen.108115>
- 12 Hidalgo M N, Gallego A J, Gil P & Pertusa A, Two-Stage Convolutional Neural Network for Ship and Spill Detection Using SLAR Images, *IEEE Trans Geosci Remote Sens*, 56 (9) (2018) 5217–5230. <https://doi.org/10.1109/TGRS.2018.2812619>
- 13 Fingas M & Brown C, Review of oil spill remote sensing, *Spill Sci Technol Bull*, 4 (4) (1997) 199–208. [https://doi.org/10.1016/S1353-2561\(98\)00023-1](https://doi.org/10.1016/S1353-2561(98)00023-1)
- 14 Espedal H A & Johannessen O M, Cover: Detection of oil spills near offshore installations using synthetic aperture radar (SAR), *Int J Remote Sens*, 21 (11) (2000) 2141–2144. <https://doi.org/10.1080/01431160050029468>
- 15 Kapustin I A, Shomina O V, Ermoshkin A V, Bogatov N A, Kupaev A V, *et al.*, On Capabilities of Tracking Marine Surface Currents Using Artificial Film Slicks, *Remote Sens*, 11 (7) (2019) 840. <https://doi.org/10.3390/rs11070840>
- 16 Solberg A H S, Storvik G, Solberg R & Volden E, Automatic Detection of Oil Spills in ERS SAR Images, *IEEE Trans Geosci Remote Sens*, 37 (4) (1999) 1916–1924. <https://doi.org/10.1109/36.774704>
- 17 Frate F D, Petrocchi A, Lichtenegger J & Calabresi G, Neural Networks for Oil Spill Detection Using ERS-SAR Data, *IEEE Trans Geosci Remote Sens*, 38 (5) (2000) 2282–2287. <https://doi.org/10.1109/36.868885>
- 18 Fiscella B, Giancaspro A, Nirchio F, Pavese P & Trivero P, Oil Spill Detection using marine SAR images, *Int J Remote Sens*, 21 (18) (2000) 3561–3566. <https://doi.org/10.1080/014311600750037589>
- 19 Kanaa T F N, Tonye E, Mercier G, Onana V P & Rudant J P, Multiscale Segmentation of Oil Slick in SAR Images Based on Morphological Pyramid, *Proc 2004 Envisat & ERS Symposium* (Salzburg, Austria), 2004, pp. 6–10.
- 20 Nirchio F, Sorgente M, Giancaspro A, Biamino W, Parisato E, *et al.*, Automatic detection of oil spills from SAR images, *Int J Remote Sens*, 26 (6) (2005) 1157–1174. <https://doi.org/10.1080/01431160512331326558>
- 21 Karathanassi V, Topouzelis K, Pavlakis P & Rokos D, An object-oriented methodology to detect oil spills, *Int J Remote Sens*, 27 (23) (2006) 5235–5251. <https://doi.org/10.1080/01431160600693575>
- 22 Topouzelis K, Karathanassi V, Pavlakis P & Rokos D, Detection and discrimination between oil spills and look-alike phenomena through neural networks, *ISPRS J Photogramm Remote Sens*, 62 (4) (2007) 264–270. <https://doi.org/10.1016/j.isprsjprs.2007.05.003>
- 23 Stathakis D, Topouzelis K & Karathanassi V, *SPIE Remote Sensing* (Stockholm, Sweden), 2006, pp. 342–350. <https://doi.org/10.1117/12.688149>
- 24 Brekke C & Solberg A, Feature Extraction for Oil Spill Detection Based on SAR Images, In: *Image Analysis. SCIA 2005. Lecture Notes in Computer Science*, Vol 3540, edited by Kalviainen H, Parkkinen J & Kaarna A, (Springer, Berlin, Heidelberg,) 2005, pp. 75–84. [https://doi.org/10.1007/11499145\\_9](https://doi.org/10.1007/11499145_9)
- 25 Huang B, Li H & Huang X, A level set method for oil slick segmentation in SAR images, *Int J Remote Sens*, 26 (6) (2005) 1145–1156. <https://doi.org/10.1080/01431160512331326747>
- 26 Topouzelis K & Psyllos A, Oil spill feature selection and classification using decision tree forest on SAR image data, *ISPRS J Photogramm Remote Sens*, 68 (2012) 135–143. <https://doi.org/10.1016/j.isprsjprs.2012.01.005>
- 27 Singha S, Bellerby T & Trieschmann O, Satellite Oil Spill Detection Using Artificial Neural Networks, *IEEE J Sel Top Appl Earth Obs Remote Sens*, 6 (6) (2013) 2355–2363. <https://doi.org/10.1109/JSTARS.2013.2251864>
- 28 Guo H, Wu D & An J, Discrimination of Oil Slicks and Look-alikes in Polarimetric SAR Images Using CNN, *Sensors*, 17 (8) (2017) 1837. <https://doi.org/10.3390/s17081837>
- 29 Gallego A J, Gil P, Pertusa A & Fisher R B, Segmentation of Oil Spills on Side-Looking Airborne Radar Imagery with Autoencoders, *Sensors*, 18 (3) (2018) 797. <https://doi.org/10.3390/s18030797>
- 30 Krestenitis M, Orfanidis G, Ioannidis K, Avgerinakis K, Vrochidis S, *et al.*, Oil Spill Identification from Satellite Images Using Deep Neural Networks, *Remote Sens*, 11 (15) (2019) 1–22. <https://doi.org/10.3390/rs11151762>
- 31 Krestenitis M, Orfanidis G, Ioannidis K, Avgerinakis K, Vrochidis S, *et al.*, Early Identification of Oil Spills in Satellite Images Using Deep CNNs, Paper presented at the *Proc 25<sup>th</sup> International Conference on Multi Media Modeling, MMM* (Thessaloniki, Greece), 2019, pp. 424–435. [https://doi.org/10.1007/978-3-030-05710-7\\_35](https://doi.org/10.1007/978-3-030-05710-7_35)
- 32 Cantorna D, Dafonte C, Iglesias A & Arcay B, Oil spill segmentation in SAR images using convolutional neural networks. A comparative analysis with clustering and logistic regression algorithms, *Appl Soft Comput*, 84 (2019) 105716. <https://doi.org/10.1016/j.asoc.2019.105716>
- 33 Zeng K & Wang Y, A Deep Convolutional Neural Network for Oil Spill Detection from Spaceborne SAR Images, *Remote Sens*, 12 (6) (2020) 1015. <https://doi.org/10.3390/rs12061015>
- 34 Patterson J & Gibson A, *Deep Learning A Practitioner's Approach*, 1<sup>st</sup> Edn, (O'Reilly Media Inc, Sebastopol), 2017, pp. 530.
- 35 Alzubaidi L, Zhang J, Humaidi A J, Dujaili A A, Duan Y, *et al.*, Review of deep learning: concepts, CNN architectures, challenges, applications, future directions, *J Big Data*, 8 (53) (2021). <https://doi.org/10.1186/s40537-021-00444-8>
- 36 Goodfellow I, Bengio Y & Courville A, *Deep Learning*, Illustrated Edn, (The MIT Press London, England), 2016, pp. 800.
- 37 Srivastava N, Hinton G & Krizhevsky A, Dropout: A simple way to prevent neural networks from overfitting, *J Mach Learn Res*, 15 (2014) 1929–1958.
- 38 Song D, Zhen Z, Wang B, Li X, Gao L, *et al.*, A Novel Marine Oil Spillage Identification Scheme Based on Convolution Neural Network Feature Extraction From Fully Polarimetric SAR Imagery, *IEEE Access*, 8 (2020) 59801–59820. <https://doi.org/10.1109/ACCESS.2020.2979219>