OIL SPILL DETECTION USING SEGMENTATION

Authors: Harshini Regonda (2156493), Samaikhya Choppadandi (2145011), Rushil Daivala (2150299), Ramalinga Reddy (2100676), Gurram Yashwanth (2145549)

University of Houston

Table of Contents

ABSTRACT

Large tankers, ships, and pipeline cracks pour oil onto sea surfaces, causing significant damage and devastation to the maritime ecosystem. Target scenarios, such as sea and land surfaces, ships, oil spills, and look-alikes, are captured using synthetic aperture radars (SAR). Oil spill detection and segmentation using SAR pictures are critical for leak cleaning and environmental protection. Based on a highly unbalanced dataset, this research introduces a two-stage deep-learning architecture for identifying oil spill events. The first stage uses a new 23-layer Convolutional Neural Network to classify patches based on the region of oil spill pixels. The second stage uses a ten-layer U-Net structure to accomplish semantic segmentation. To account for the oil spill representation in the patches, the dice loss is minimized. The findings of this investigation are quite encouraging, as they show improvement in unbalanced dataset and focuses mostly on identification and segmentation.

1. INTRODUCTION

Crude oil is a fossil fuel that is used to generate a variety of fuels and products. It is the liquid leftovers of ancient plants and animals. Oil is found in reservoirs beneath the ground or beneath the ocean's surface, where oil droplets reside in "pores" or holes in the rock. Following the drilling and pumping of crude oil, oil corporations transport it to refineries through pipelines, ships, trucks, or trains. During transportation there could be some accidents occurring.

Oil spills happen more frequently than you might imagine, and they can occur in a variety of ways. Each year, thousands of oil spills occur in US seas. The majority of these spills are minor, such as fueling a ship. However, these spills can still cause harm, particularly if they occur in sensitive areas such as beaches, mangroves, and wetlands. Large-scale oil spills are huge, life-threatening events. Pipelines break, large oil tanker ships sink, and drilling operations go wrong, and these things happen. After a significant oil spill, the effects on ecosystems and economies can last for decades.

Oil spill can occur anytime, during drilling, transportation, or usage. NOAA experts may be called in when oil spills occur in the ocean, the Great Lakes, on the shore, or in rivers that drain into these coastal waterways. The aim of the Office of Response and Restoration is to come up with scientific ways to preserve the coasts free of oil, chemicals, and marine debris. The location of the oil spill, the kind of plants, animals, and ecosystems present there, as well as the volume and type of oil spilt, all play a role in how much damage an oil spill causes. Oil spills wreak havoc on marine life. Oil spill response frequently includes wildlife recovery, cleaning, and rehabilitation. Wildlife, on the other hand, is difficult to locate and capture, oil spills can occur over large areas, and certain creatures are too large to be recovered. Unfortunately, rescuing all species affected by oil spills is unrealistic.

SAR (Synthetic Aperture Radar) is a type of radar that may be deployed on planes or satellites to gather images of sea and land surfaces. The SAR's sensors give out radio waves, which are reflected off the surfaces and used to create a visual depiction of the target surface. Sea, land surfaces, oil spills, ships, and look-alikes may all be captured in SAR photos. Low-speed wind zones, sea wave shadows, and grease ice are only some of the environmental phenomena that look-alikes could portray. In SAR photos, radio waves reflected by oil spills or look-alikes appear as dark or black dots, making it difficult to distinguish oil spills from other look-alikes.

i. Literature review

(Shamsudeen Temitope Yekeen, Abdul-Lateef Balogun) Evaluating the various remote sensing techniques used to identify oil spills in marine environment and reviewing in detail, the limitations of various telemetry devices, multiple oil spill detection models and upcoming computer vision software, the author recapitulates the obstacles and opportunities of using different ML and DL models. Concluding that every model has some or other imperfections the author states the numerous reasons for the inaccuracies. To overcome the inaccuracies, it is imperative to develop new models that resolve the limitations.

(Peng Liu, Ying Li, Bingxin Liu, Peng Chen, Jin Xu) Texture analysis to detect oil spills, where a texture index computed from contrast, correlation, entropy, and energy to indicate the location of the spill, was proposed by the author. The radar images of oil spill accident on 21st July 2010 at Dalian were classified using ML models like, SVM, KNN, ensemble learning, and Linear discriminant analysis. Adaptive thresholding method used for fine measurements. Oils spills were detected semi-automatically by selecting training data on the proposed texture index image, no manually set thresholds were required according to the proposed method. The final accuracy of the proposed method was 80%.

(ShamsudeenTemitope Yekeen, Abdul-LateefBalogun, Khamaruzaman B.Wan Yusof) The author proposes an advanced method for precise recognition and segmentation of oil spill detection, learning from the texture and shape. A novel DL model was developed using Mask R-CNN for oil spill detections and segmentation. The model has an advanced two stage methodological approach. The first stage begins with image pre-processing constituting dB linear transformation, single layer spackle filter, radiometric calibration, and rescaling. The Mask R-CNN model was then developed and trained by ResNet 101 backbone with Feature Pyramid Network (FPN) architecture and COCO dataset. The Mask R-CNN model outperforms other models with lowest validation loss from testing (0.05922).

ii. Business/Analytics Problem

The water contamination by oil and its derivatives is a worldwide concern. Oil spill accidents can cause severe ecological disasters; hence, the timely and effective detection of oil spills on the marine surface is of great significance. The frequency of marine oil spills has increased in recent years. The growing exploitation of marine oil and continuous increase in accidents during marine crude oil transportation has caused tremendous damage to the marine ecological environment. It is one of the major causes of water pollution. Using synthetic aperture radar (SAR) images to monitor marine oil spills can help control the spread of oil spill pollution over time and reduce the economic losses and environmental pollution caused by such spills.

iii. Objective

This research focuses on the use of deep learning algorithms for oil spills classification and segmentation. UNet is a convolutional neural network, originally proposed for biomedical image segmentation and modified for the discrimination of oil spills and look-alikes. The model is trained on a publicly available benchmark oil spill detection dataset of synthetic aperture radar (SAR) images.

iv. Impact and Value

The aerial view segmentation system helps in identifying the accident area at a faster pace, minimizing human intervention, helps in tracking the pipelines covering a wider view resulting in cost and time efficiency. The short-term economic damages for an organization can be prevented if the oil spills are detected.

2. DATA

Satellite images of oil spills near marine coastal areas, oil spills in deep oceans. The data was collected from different resources such as Statista, Nasa satellite images. The images vary in sizes and pixels. The images are adjusted according to the requirement.

i. Dataset background and quality

The first process is data collection, which was done using Statista, which comprises 84 photos in JPEG and XML formats. There are 52 images of oil spills and 32 images of non-oil spills among the 84 total. Oil spill images were split into 70:30 ratio for training and testing.

ii. Data processing, wrangling, and EDA

The data is loaded in the second step. We need to resize all the images to the same size since all the image data should be the same size after passing through any machine or deep learning model. After resizing, the image labels are also changed. The grayscale images of width and height (96*96*1) are converted to RGB images of size 96*96*3 in this project.

The data from the csv files is sent into EDA in the next phase, with class values of 1 for oil spills and 0 for non-oil spills. The sizes for oil spill photographs are taken from the top down, while non-oil spill images are (1*96). In the next phase, the missing values are dealt with accordingly.

Keras works with images in batches. The number of samples (or images) you have is represented by the first dimension. When you load a single image, you obtain its shape, which is in the form (size1, size2, channels). So, to create a batch of images you need an additional dimension like (samples, size1, size2, channels). The preprocess input function is used to convert your image into the format required by the model.

3. METHODOLOGY

i. Workflow

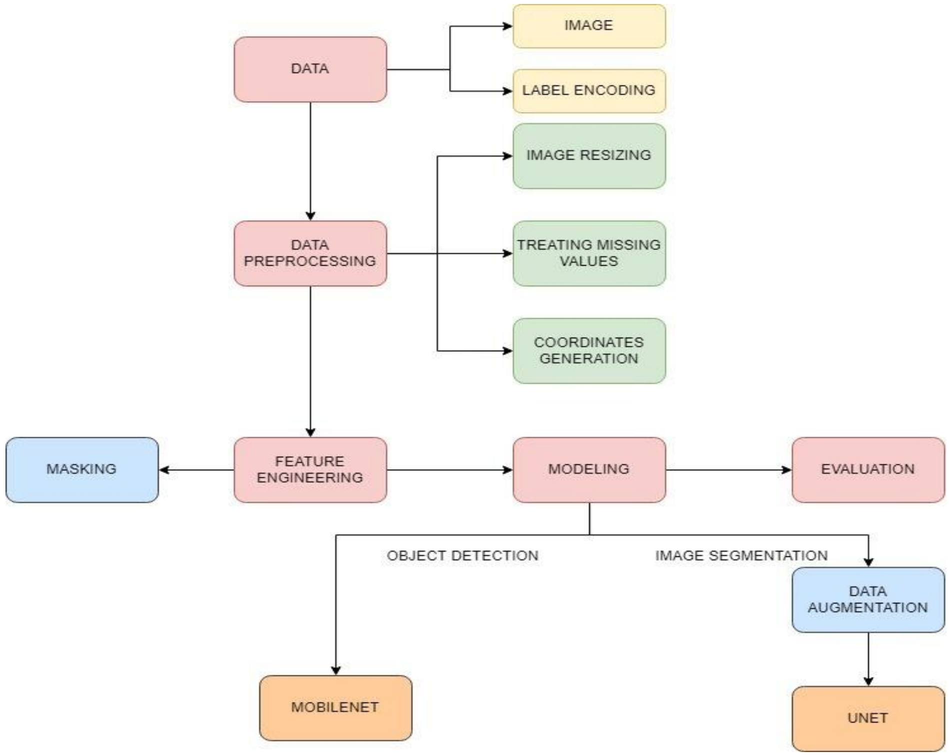


Fig 1: Workflow of the project

The workflow of the project involves other steps after the data preparation. Once the data is processed, we apply feature extraction i.e., Masking to enhance the contours required for the image. The above prepared data is fed into the models to obtain various results which are described further.

ii. Methods description

Object Recognition

Object recognition concerns the identification of an object as a specific entity (i.e., semantic recognition) or the ability to tell that one has seen the object before (i.e., episodic recognition). In this project for the object recognition, we are using Mobilenetv2 architecture for image recognition which is state of art developed by google.

Object Segmentation

Semantic segmentation is the process of providing a category label to each pixel in an image to separate and detect certain categories while also finely labeling their boundaries. In addition to identifying each category, instance segmentation assigns various labels to individual instances of the same object type. The U-Net is trained for image segmentation in the project.

iii. Feature Engineering

Image labeling is a way to identify all the entities that are connected to, and present within an image. The images can have multiple entities present within it, we concentrate only on the label that is required and the remaining background is removed. The required region is mapped by drawing polygon over the contours and moved onto a plain background which is called masking. Dataset is considered for masking to create labels.

Fig 2: Example of masking of images

iv. Modeling

Mobilenetv1:

MobileNetv2 builds upon the ideas from MobileNetv1, using depth wise separable convolution as efficient building blocks. MobileNetv1 is a CNN architecture that is both efficient and portable, and it is employed in real-world applications. MobileNets typically use depth wise separable convolutions to develop lighter models instead of the usual convolutions used in previous architectures.

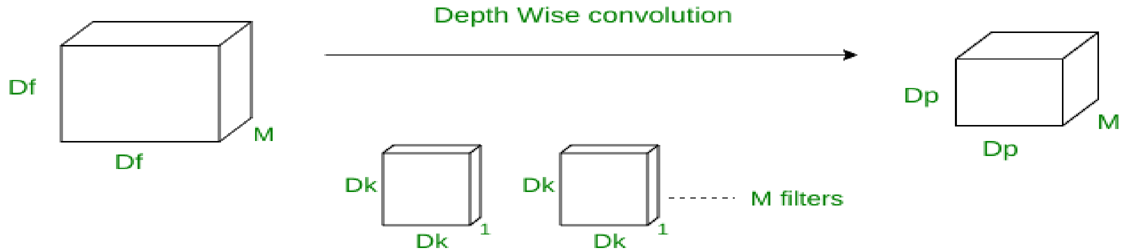
Depth-wise separable convolutions are broken down into two operations:

a. Depth wise convolutions
b. Point wise convolutions

Depth-wise convolution:

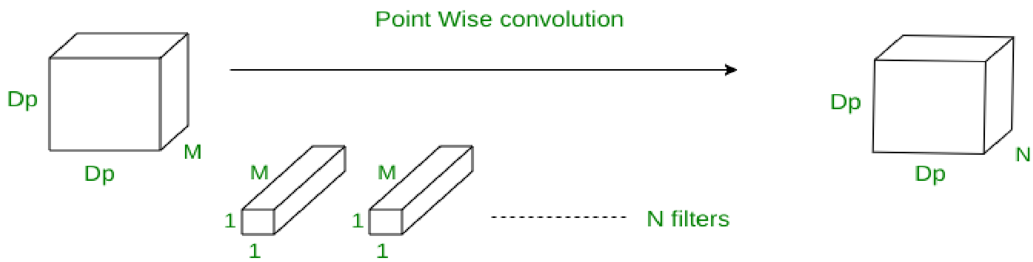
In depth-wise operation, the convolution is applied to a single channel at a time, in contrast to standard CNNs where it is done for every M channels. So here the filters/kernels are of size Dk x

Dk x 1. Assuming there are M channels in the input data, then M such filters are required. The output is of size Dp x Dp x M.



Point-wise convolution:

In point-wise operation, a 1×1 convolution operation is applied on the M channels. So, the filter size for this operation will be 1 x 1 x M. Say we use N such filters, the output size becomes Dp x Dp x N.



Mobilenetv2:

In contrast to standard residual models, the MobileNetV2 design is built on an inverted residual structure, in which the residual block's input and output are thin bottleneck layers. Instead of using merely the deep separable convolution block, it employs the bottleneck residual block. The convolution network has three layers:

a. Expansion Layer
b. Depth wise convolution layer
c. Projection Layer

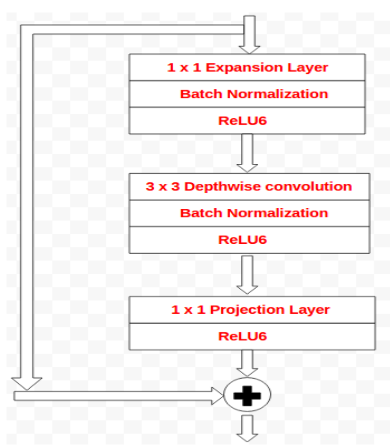


Fig 3:MobileNet Layers

A 1 x 1 expansion layer is the first layer. It expands the amount of data that passes through it (by increasing the number of channels). Its function is the opposite of that of the projection layer. Based on the expansion factor, the data is expanded. This is a hyperparameter that can be discovered through various architecture trade-offs. The expansion factor is set to 6 by default.

The depth-wise convolution layer is the second layer. (Discussed above in mobilenetv1)In MobileNetv1, pointwise convolution either keeps or doubles the number of channels. However, in this case, 1 x 1 convolution reduces the number of channels. it is known as projection layer. Because it reduces the quantity of data that travels through it, this layer is also known as the bottleneck layer.

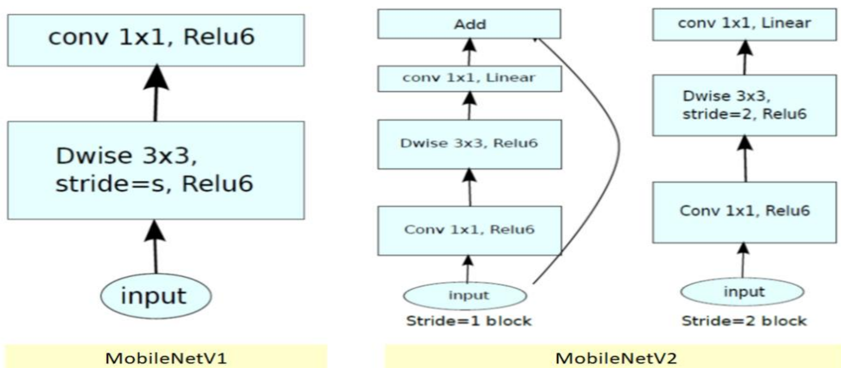


Fig 4: Differences between the architectures

Architecture:

In the mobilenetv1 we use only 1 block whereas in mobilenetv2 we use 2 blocks.

An inverted bottleneck block will not have a residual connection if stride=2 is used for depth wise convolution.

Input	Operator	Output
$h \times w \times k$	1x1 conv2d , ReLU6	$h \times w \times (tk)$
$h \times w \times tk$	3x3 dwise s=s, ReLU6	$\frac{h}{s} \times \frac{w}{s} \times (tk)$
$\frac{h}{s} \times \frac{w}{s} \times tk$	linear 1x1 conv2d	$\frac{h}{s} \times \frac{w}{s} \times k'$

The Expansion factor used in most layers is 6. The output has 64*6=384 channels if the input has 64 channels.

Input	Operator	t	c	n	s
$224^2 \times 3$	conv2d	-	32	1	2
$112^2 \times 32$	bottleneck	1	16	1	1
$112^2 \times 16$	bottleneck	6	24	2	2
$56^2 \times 24$	bottleneck	6	32	3	2
$28^2 \times 32$	bottleneck	6	64	4	2
$14^2 \times 64$	bottleneck	6	96	3	1
$14^2 \times 96$	bottleneck	6	160	3	2
$7^2 \times 160$	bottleneck	6	320	1	1
$7^2 \times 320$	conv2d 1x1	-	1280	1	1
$7^2 \times 1280$	avgpool 7x7	-	-	1	-
$1 \times 1 \times 1280$	conv2d 1x1	-	k	-	

The expansion factor is ‘t’, and the number of output channels is ‘c’. The repeating number is ‘n,’ and the stride is ‘s’.

The mobilenetv2 architecture has 25 layers in which 23 are convolution layers, 1 fully convoluted layer and another 1 resizing layer. The model summary is shown below:

Once the model is created the images are then sent into training. The train-test is set to 70:30 ratio. The optimizing algorithm used is Adam optimizer. The learning rate is set to 0.3 and the activation function is SoftMax. The loss function which we are going to use is focal loss binary. The number of iterations (epochs) used is 3 and the batch size used is 3. The metrics for evaluation used is accuracy. Then the images are considered for testing and the class value is predicted.

Adam Optimizer:

Adaptive Moment Estimation is a technique for optimizing gradient descent algorithms. When working with the problems involving a lot of data or parameters, the method is quite efficient and takes minimal memory. It's essentially a hybrid of the 'gradient descent with momentum' and the 'RMSP' algorithms.

Adam Configuration Parameters:

Learning rate or alpha: Also referred to as the step size. The percentage by which weights are updated (e.g., 0.001). Larger values (e.g., 0.3) result in faster initial learning before the rate is updated. Smaller values (e.g., 1.0E-5) slow down learning even during training.

Adam's hyperparameters 1(beta_1) and 2(beta_2) are initial decay rates used in estimating the first and second moments of the gradient, which are multiplied by themselves at the end of each training step (batch).

Epsilon: It is a very small number to prevent any division by zero in the implementation (e.g., 10E-8).

Focal Loss:

Focal Loss is an improved version of Cross-Entropy Loss that attempts to handle the class imbalance problem by assigning more weights to severe or slightly misclassified examples and down weighting simple examples.

$$FL(p_t) = -\alpha_t(1 - p_t)^{\gamma} \log(p_t).$$

The Focal Loss adds a factor (1−pt) γ to the standard cross entropy criterion. Setting γ>0 reduces the relative loss for well-classified examples (pt>.5), putting more focus on hard, misclassified examples. Here there is tunable *focusing* parameter γ≥0.

Binary Focal loss:

Binary Focal loss function generalizes binary cross-entropy by introducing a hyperparameter called the focusing parameter that allows hard-to-classify examples to be penalized more heavily relative to easy-to-classify examples.

U-Net Architecture:

U-Net is one of the standard CNN architectures for image segmentation tasks. It is considered as a best network for fast and precise segmentation of images. The network is based on a fully convolutional layers whose architecture was modified and extended to work with fewer training images and yield more precise segmentation.

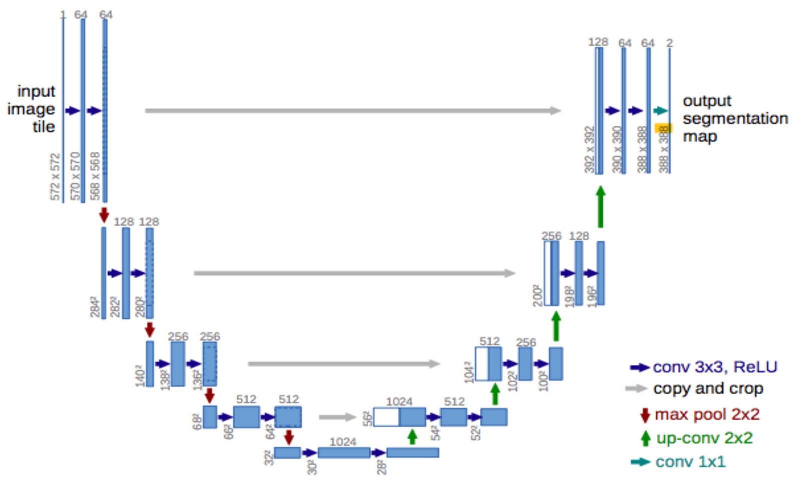


Fig 5:Architecture of Unet

U-Net architecture consists of two paths:

Encoder path: The encoder path is for feature extractions.
Decoder path: Decoder path is for localization of the objects.
There are four convolution blocks in the encoder path, each with two convolution layers: 3x3 padding followed by a ReLU (Rectified Linear Unit), 2x2 Max Pooling operator with two strides. With each successive convolution block, the number of filters doubles. Initially, there are 64 filters. Each filter helps to increase depth. There are no max pooling operators in the fifth convolution block; it then connects to the Decoder path.
There are four convolution blocks in the Decoder path, each with two 2x2 convolution layers. It has the same symmetry as the Encoder path. Each convolution increases the resolution while

decreasing the depth. The number of feature channels continues to be cut in half. The convolution layer is followed by 3x3 filters with ReLU activation. The 1x1 final convolution layer helps in precise localization by mapping features to the appropriate classes.

Once the model is ready, the images are sent for training. The learning rate is set to 0.00003 and the number of iterations is 100 and batch size 3. DICE loss function is used. The optimizer used is Adams. The metrics used are mIOU, accuracy and precision.

Jaccard's Co-efficient:

Also known as mean intersection over union. For image recognition and segmentation, the metric used for evaluation is mIOU which is,

$$IoU = \frac{Intersection}{Union} = \frac{\#TP}{\#TP + \#FP + \#FN}$$

mIOU is mean of all the classes IOU. This is a useful indicator of how effectively an image segmentation model works across all the classes it should detect.

v. Training and Testing

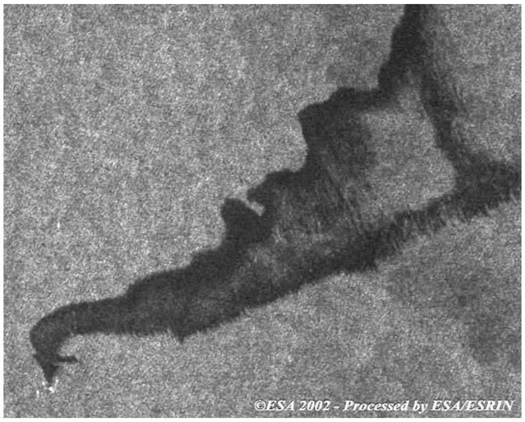
Once the models are built with the requirements, the models are trained with 70% of the dataset and tested with 30% of the dataset. This is a standard ratio of division of dataset. Once the training is done the images are sent for testing and respective metrics are evaluated.

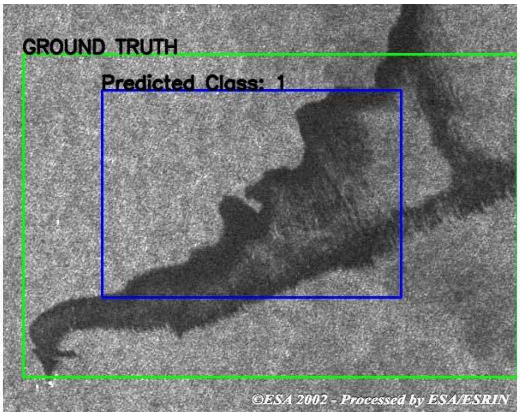
4. RESULTS AND DISCUSSION

i. Models Evaluation

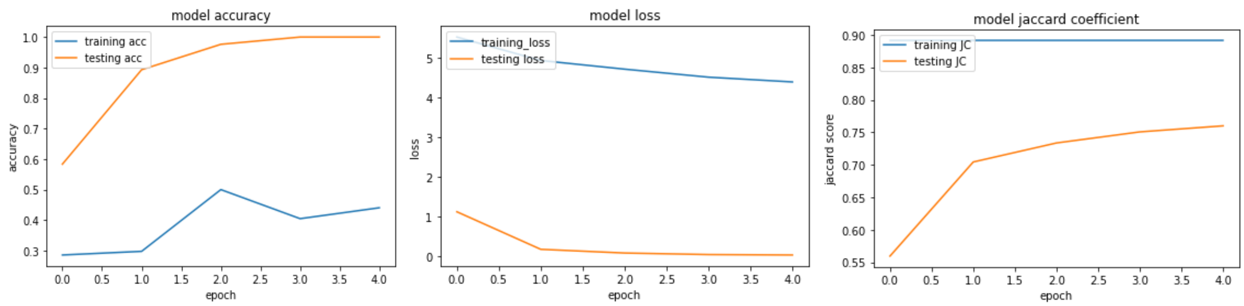
The metrics evaluated here are accuracy, loss and mIOU.

The mobilenetV2 model outputs are as follows:

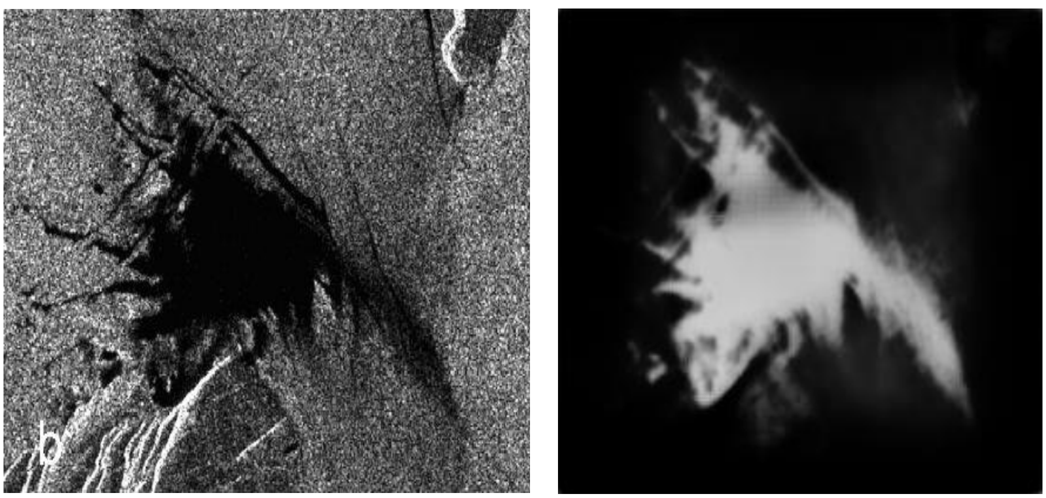




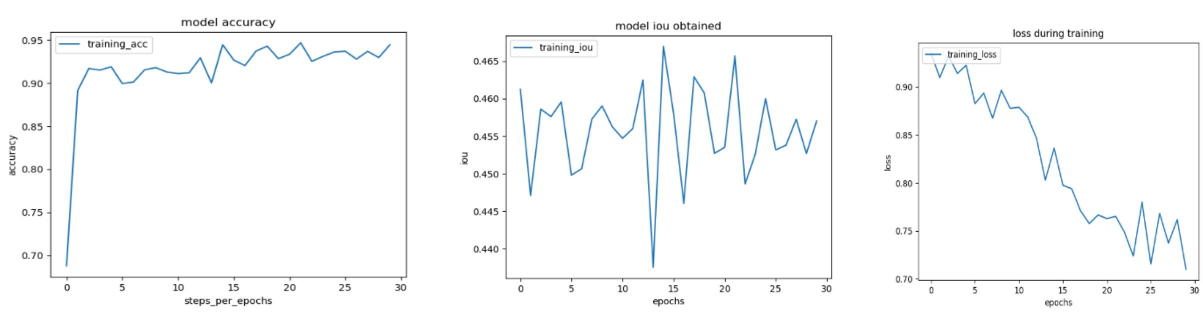
The first model helps in recognizing oil spills with an accuracy of 92% and with a loss of 4.3 with 5 epochs. The model successfully predicted the classification by identifying regions around the oil spill.



The output of the segmentation is as follows:



The second model segmented the images with an accuracy of 94.4%, mIOU of 0.701 with 30 epochs, steps_per_epoch = 10. The output has a loss of 33.16%. The output is a black and white image with labelled and unlabeled data.



5. CONCLUSION AND RECOMMENDATIONS

Oil spills have enormous impact on economy, employment, and transportation. Unfortunately, accidents occur, and spills reoccur more frequently. With the help of deep learning, this helps in faster identification of the oil spill and also in identification of the origin of the spill which helps in controlling the spread of spill and faster damage control is achieved.

6. ACKNOWLEDGMENTS

Without the great help of my Professor, Dr. Mohamed Mohamed, this work, and the research underlying it would not have been possible. From our initial meeting with the machine learning algorithm to the final draft of this paper, his passion, competence, and meticulous attention to detail have been an inspiration and kept our work on track. Our interest in incorporating machine learning algorithms into the petroleum industry has grown because of the Big Data Analytics for Petroleum Engineering class. We are also appreciative for the helpful feedback from our class peer reviewers. The generosity and competence of each one of you has improved this study in countless ways and saved us from numerous mistakes; those that inevitably remain are only our responsibility.

7. REFERENCES

1) TY Shamsudeen et al.., 2020 - Advances in Remote Sensing Technology, Machine Learning and Deep Learning for Marine Oil Spill Detection, Prediction and Vulnerability Assessment
2) P Liu et al..., 2019 - Semi-automatic oil spill detection on x-band marine radar images using texture analysis, machine learning, and adaptive thresholding Google Scholar
3) ST Yekeen et al…, 2020 - A novel deep learning instance segmentation model for automated marine oil spill detection Google Scholar
4) Diederik P. Kingma and Jimmy Ba ,2017 - Adam: A Method for Stochastic Optimization
5) Andrew G. Howard et al…, 2017 - MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications
6) Mark Sandler et al…, 2019 - MobileNetV2: Inverted Residuals and Linear Bottlenecks
7) Olaf Ronneberger et al…, 2015 - U-Net: Convolutional Networks for Biomedical Image Segmentation
8) AC Bejarano et al…, 2016 - Oil spills and their impacts on sand beach invertebrate communities: A literature review
9) AA Al-Majed et al…, 2012 - A sustainable approach to controlling oil spills
10) Object Segmentation
11) M. DoloresGarza-Gil et al…, Estimating the short-term economic damages from the Prestige oil spill in the Galician fisheries and tourism