Development of a Location Invariant Crack Detection and Localisation Model (LICDAL) in Unconstrained Oil Pipeline Images Using Deep Convolution Neural Networks

Bolanle F. Oladejo oladejobola2002@gmail.com
Department of Computer Science
University of Ibadan,
Ibadan, 200284, Nigeria.

Chigozilai B. Kejeh kebochig@gmail.com
Department of Computer Science
University of Ibadan,
Ibadan, 200284, Nigeria.

Abstract

Computer vision (CV) -based techniques are being deployed to solve the problem of Crack Detection in metallic and concrete surfaces. This is because the Human-oriented inspections being used have drawbacks in the area of cost and manpower. One of the deployed CV techniques is the Deep Convolutional Neural Network (DCNN). Existing DCNN based crack detection models have a challenge of performing poorly when tested on images taken at a different location from the training images, hence crack localization is required. Thus, this research develops a location invariant crack detection and localization (LICDAL) model in unconstrained oil pipeline images using DCNN. LICDAL is developed by applying transfer learning on the Faster Region based - CNN (Faster R-CNN). The model is made location invariant by gathering images of cracked oil pipeline from various locations. The collected images are split into a 70%:30% ratio for training and testing set. LICDAL is evaluated using the mean Average Precision (mAP). The results on testing LICDAL shows the detected and localised cracks with a mAP of 97.3% on a set of 10 new test images taken from different locations; the highest Average Precision at 99% and the lowest Average Precision at 86%. The performance of LICDAL is compared to an existing crack detection model which detects cracks alone. LICDAL adequately localizes the detected cracks, thus improving crack identification. Secondly, there is no drastic reduction in performance for the test images taken at different locations from the training images, thus making LICDAL location invariant.

Keywords: Deep Learning, Convolutional Neural Network, Transfer Learning, Crack Detection.

1. INTRODUCTION

The utilization of oil pipelines for the transportation of oil products is of no little significance in Oil production. Though this method is more effective and efficient for oil product transportation, it still has its dangerous disadvantages in the form of “oil spills”. [1] defines Oil spill as an unintentional release of liquid petroleum hydrocarbon into the environment as a result of human activities which are usually caused by accidents involving pipelines, refineries, among others. As [2] notes, the environmental consequences of oil pollution on inhabitants are numerous; oil spills over time have turned productive land mass into badlands. Not only does it affect land, but as pointed out by [3], the total spillage of petroleum in seas, rivers and oceans through human activities is estimated 0.7 -1.7 million [3]. He also pointed out that oil spills could lead to the total destruction of ecosystems. A couple of Computer Vison and Image Processing techniques have been adopted in automatic detection of defects in structures, including cracks. [4] opined that, Computer vision and Image processing techniques are being utilized to visually inspect structures surface defects, including cracks, and the importance of image processing for visual inspection has been on the increase in various fields. Since visual inspection can examine a wide range of structure surfaces in one shot by taking a picture of the desired areas, it will be of a high advantage as cost and manpower could be greatly reduced while inspecting. An aspect of Machine Learning that has seen more research in recent times is Deep Neural Network also known as Deep Learning, which seeks to learn a hierarchy of features at different levels from an input dataset. The edge Deep learning has over other algorithms is its ability to automatically learn a hierarchy of features from a set of unprocessed input data without the need for feature engineering as is the norm with most Machine Learning algorithms; it does this by employing deep Neural Network architectures [5]. Unlike shallow Neural Networks with few hidden layers (say 1 or 2 layers), Deep Learning methods utilize Neural Networks with many hidden layers or Deep architectures which allows for automatic learning of features on multiple levels. This is why Deep Learning is used in Computer Vision and Image processing tasks, and has proven to be more efficient than the traditional computer vision and image processing techniques.

A lot of research on pavement and concrete crack detection using Deep Convolutional Neural Network has been carried out, however some areas still remain quite challenging. To the best of our knowledge there are hardly literature which has dealt with localizing the detected crack using DCNNs, that is, creating a bounding box around the region of the crack for easier detection. We propose the use of Deep Convolutional Neural Networks (DCNNs) and Transfer learning in detecting and localising cracks in unconstrained images of oil pipelines. Localisation of an object in an image means to draw a bounding box around that object once detected in the image. Unconstrained images refers to images obtained under diverse uncontrolled imaging condition including different lighting, texture, rotation, shadow changes etc. Transfer learning implies using a working deep learning model pre-trained on image datasets with lots of images, and transfer its learnt weights to new classification problems. On the other hand, painstakingly building and training a DCNN from scratch can be time and resource consuming, that is if the resources, in terms of training images, are sufficient. Proper fine-tuning of these pre-trained DCNN models have been seen to outperform DCCNs trained from scratch as was stated by [6] when tested on medical images.

The rest of the paper is arranged as follows: section 2 gives a review of the crack detection techniques in use, ranging from Human-based techniques to Computer-vision based techniques; the latter part of this section gives a quick introduction on DCNN, Object detection, and the idea of transfer learning on DCNNs. Section 3 explains the proposed methodology adapted in this paper. In section 4, we discuss the Data collection and implementation. In section 5, the training and test result of the model are discussed and a comparative analysis with a DCNN-based method is made. The article concludes with a summary and recommendations for future work in section 6.

2. RELATED WORKS

Generally, cracks are detected on surfaces via an all-encompassing method called Nondestructive testing (NDT), which is the branch of engineering concerned with non- contact methods of detecting and evaluating defects in [7]. The reason NDT is mostly used is because it is non-destructive on the material or structure being tested. NDT ranges from simple techniques such as visual examination of surfaces, through the well- established methods of radiography, ultrasonic testing, magnetic particle crack detection, to new and very specialized methods [8]. NDT is classified into two categories as mentioned [8]: Non-visual and Visual NDT methods. The Non-visual NDT methods involves the travelling of waves (sound, current, etc.) into the object and the feedback from these waves determine the state of the object. Some techniques under the Non-visual NDT methods are: Ultrasonic inspection, Eddy current testing and Acoustic emission monitoring, etc. The visual NDT methods involves visual inspection and are much easier to perform, inexpensive and does not require the use of special equipment. Some techniques under this method are: Closed circuit television (CCTV) inspection, Pipe scanner and evaluation technology (PSET) and Laser-based scanning systems. Although these aforementioned techniques are effective, a number of Computer Vison and Image Processing techniques have been adopted in automatically detecting defects in structures, including cracks. These techniques basically involve automatically recovering useful information about a scene from its twodimensional projections [8]. Since visual inspection can examine a wide range of structure surfaces (in some cases, remotely) in one shot by taking a picture of the desired areas, it will be of a high advantage as cost and man power can be greatly reduced while inspecting [4].

As reported in literature, the use of Image processing and Computer vision techniques in automatic crack detection has been progressive and impressive over the years. [9] adapted and implemented an efficient algorithm for detecting crack patterns in pipeline images. The automated method is divided into three steps, namely, contrast enhancement, morphological treatment and curvature evaluation in the cross- direction and finally the alternating filters that produce the final segmented binary crack map. Their proposed evaluation scheme adequately estimated the performance of this algorithm in an absolute way and results showed that 91% of cracks were detected. [4] proposed the use of Percolation model and Edge information for crack detection on concrete surfaces. They were able to detect connected cracks adequately and this method outperformed a previous method where Wavelet transforms were used, which detected more noise alongside the cracks. Gabor filters were used by [10] to detect pavement cracks; they were able to detect diverse types of cracks with up to 95% precision. In [11], a method to automatically detect cracks in images of concrete through segmentation using fuzzy c-means clustering and multiple noise reduction was proposed. They used a 3 stage model to first perform image segmentation using fuzzy c-means, then mask filters of different sizes were used to remove noise leaving only the cracks. This method outperformed the Sobel detection segmentation, however, it still contains lots of noise.

The above-mentioned methods utilize Image processing techniques to manipulate images and extract crack features. However, the challenge with these Image processing techniques is its inability to perform optimally under unconstrained imaging conditions like lighting and shadow change. In a bid to solve this challenge, Convolutional Neural Network (CNN), which is a Computer Vision based method, is used to automatically detect cracks without having to hand engineer the image features to look for. Hence, Crack detection using CNN has proven to be effective. [12] used a 3 layer CNN to build a classifier to detect concrete cracks from unconstrained images. In order to properly train the network, a sliding window technique was used to scan through the Image to detect cracks. Images were cropped in to smaller images of size 256 x 256 pixels, which is the size of the sliding window; this was used for the training and validation of the network. This method yielded good results for the accuracies in training and validation of 98.22% at the 51st epoch and 97.95% at 49th epoch, respectively. The problem with this method is its heavy computation increasing the running time which took about 1-2 days on a CPU only system; with two GPUs it took about 90 minutes until the 60th epoch. [13] proposed a 4 layer supervised deep convolutional network to classify image patches collected by a low cost smart phone. The CNN was trained on square image patches of size 5x5 pixels, and a patch whose centre is a crack pixel or is close to a crack pixel is classified as a crack. Though this method adequately detects cracks, it has a downside of classifying more pixels in the image as cracks than actually is in the original image. [14] presented a 4 layer convolutional network to detect cracks in pavements. The layers consisted of 4 convolutional layers, 4 max pooling layers, 2 fully connected layers, an auxiliary dropout layer and a Softmax layer. They trained and tested images from a certain location with the 4 layer network and added a 5th layer which was used to train and test images from another location. The 5 layer network performed better than the 4 layer network in terms of accuracy, precision and recall with a score of 91.3%, 90.7% and 92.0% respectively. However, the classifier performed poorly on test images that were taken from a different location from the training images with a score of 90.1%, 85.6% and 96.4% for the accuracy, precision and recall . [13] worked on a Convolutional neural Network which automatically detects pavement cracks in asphalt surfaces, the CNN was named CrackNet. Unlike regular CNNs, this method has no pooling layers to reduce the sizes of the images during training; the size of the input image is invariant through all the layers of the ConvNet achieving a pixel-perfect accuracy. CrackNet consists of one 1 x 1 convolutional layer, two fully connected layers and an output layer; it uses more than one million parameters. The results show a 90.13% precision. [15] proposed a method of using a pre-trained deep learning model and transfer learning to detect crack from pavement images. Using the Keras deep learning framework, they used an open sourced implementation of the VGG-16 (a 16 layer Deep ConvNet) model which has already been trained on the ImageNet database. The fully connected layer of the VGG-16 is truncated and is used as a deep feature generator which produces semantic image vectors. These image vectors were trained and tested using different machine learning techniques, of which Neural Network outperformed the others with a balanced accuracy of 87%. Also of note is the work done by [16] where they developed a dataset for road damage and trained and evaluated a damage detection and localisation model through transfer learning based on a combination of some state-of-the-art convolutional Neural Networks. They used a mixture of SSD and MobileNet to build a model that can run on a smartphone and achieved recalls and precisions over 75% with an inference time of 1.5s on a smart phone.

2.1 Deep Convolutional Neural Networks (DCNNs)

DCNNs is an effective architecture in processing visual data such as images and videos. DCNNs accept raw input data at the lowest level and transforms them by processing them through a sequence of basic computational units to obtain representations that have intrinsic values for classification in the higher layers [17]. DCCN consists of three layer types: convolutional layers, subsampling layers and fully connected layers. The Convolution layer takes the convolution of the input image with the convolution matrix and generates the output image. A convolutional layer is parametrized by the number of channels, kernel size, stride factor, border mode, and the connection table. Multiple convolutional layers are used to take into consideration the spatial dependencies among image pixels. [15]. The subsampling layer is used to make the neural network more invariant and robust. This layer is mainly to reduce the input image size in order to give the neural network more invariance and robustness. The most used method for subsampling layer in image processing tasks is max pooling as it has shown to lead to faster convergence and better generalization. The subsampling layer is frequently called max pooling layer [15]. Full connection layers are similar to the traditional feed-forward neural layer. They make the neural network fed forward into vectors with a predefined length. We could fit the vector into certain

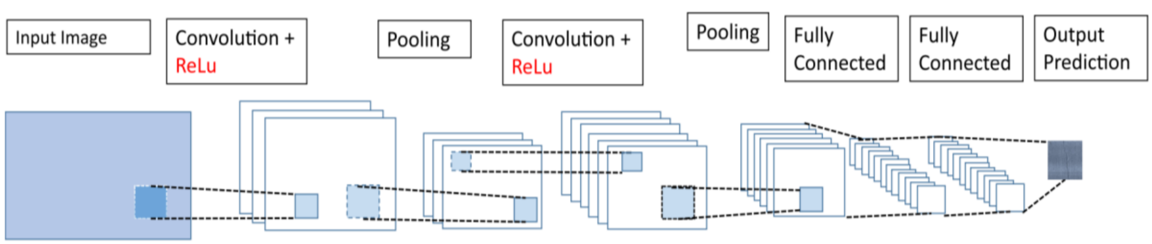

FIGURE 1: Schematic diagram of Convolutional neural network [15].

categories or take it as a representation vector for further processing. Figure 1 gives a pictorial view of a CNN.

2.2 Transfer Learning

Training a DCNN is quite tedious, as the training dataset has to be of a sufficiently large number which will be trained within a certain period of time depending on the processing power of the machine used. For some cases, gathering of sufficient data is very tasking or impossible, and training a DCNN to generalize (and not overfit) will require lots of training dataset. Where this is the case, then it is possible to take an already trained DCNN model and use it to perform the given task; this is known as Transfer Learning or Fine-tuning. The intuition is that it is cheaper and easier to use state-of-the-art DCNN models already trained on very large data and transfer their learning ability to the classification task rather training one from scratch. As shown by [6] in his work, for medical applications, the use of a pre-trained Convolutional neural network (CNN) with adequate fine-tuning outperformed or, in the worst case, performed as well as a CNN trained from scratch.

2.3 Object Detection

Object detection is aimed at locating the different objects in an image and classifying them in their different categories; this is normally done by putting a bounding box around the image for easy identification and labelling. This is different from Image classification which seeks to classify an image into a category based in the most prominent object. Object detection can be used to locate the different objects in an image, count the objects and also segment the objects from the image. DCNNs which have been widely used for Image classification are also used for Object detection. Essentially, there are three steps involved in an object detection framework:

1. An algorithm to create a number of bounding boxes to capture the objects in the images. This is referred to as Region of Interest (RoI) or Region Proposals.
2. Image classification of each of the objects selected in the region proposals based on visual features. This is where the DCNN is utilized.
3. For the final stage, overlapping boxes are combined into one using certain algorithms.

The Object detection model used for this paper is Faster R-CNN, which is discussed in the following section.

3. METHODOLOGY

This Section presents an overview of the general framework of the crack detection model. The framework consists of two major stages - the Pre-processing stage and the Crack detection model design stage. The first stage consists of processes that prepare the images for training and validation on the crack detection model while the second stage deals with designing the model. The Crack detection model design stage shall be discussed next. The pictorial view of the general framework is shown in Figure 2.

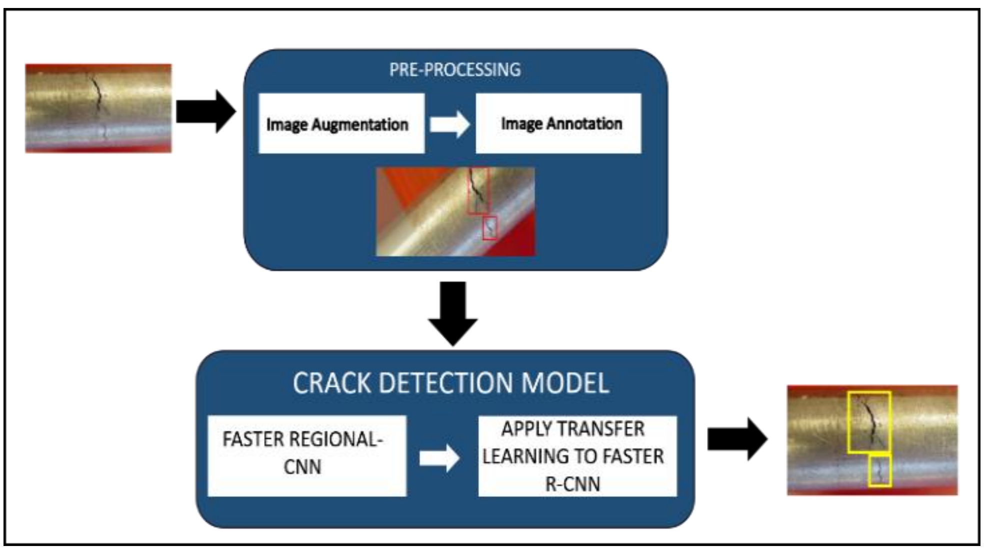

FIGURE 2: General Framework of the Crack detection model.

The Faster Regional Convolutional Neural Network (Faster R-CNN) is the Object Detection model adapted for the Crack Detection and Localisation model. Proposed by [18], Faster R-CNN is used because of its ability to process images and videos in quasi real-time while giving a high performance. In order for us to use its already learned weights for our purpose, transfer learning was applied on the model by freezing some layers of the network and changing some layers (the final layers). The Faster R-CNN comprises of two components - the Region Proposal Network (RPN), for region proposals, and the Fast R-CNN, for classification. The peculiarity of this model is that both the RPN and the Fast R-CNN share the same CNN to extract features from images. The architecture of the Faster R-CNN is shown in Figure 3.

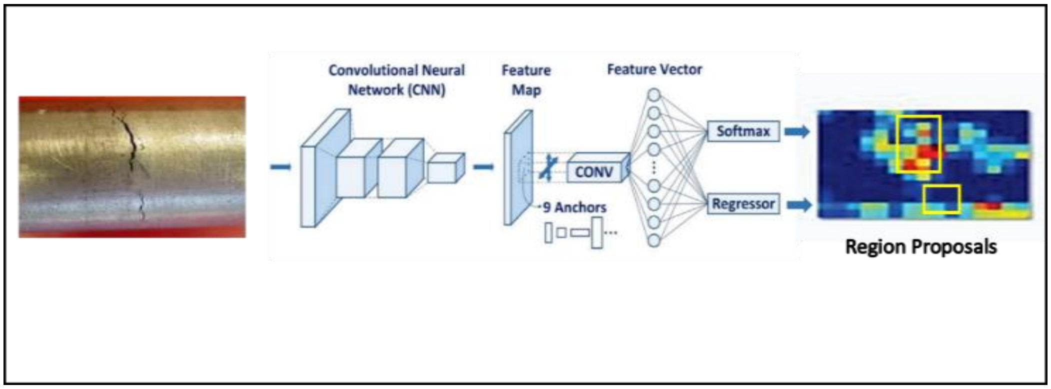

FIGURE 3: Overview of the adapted Faster R-CNN architecture [12].

3.1 Region Proposal Network (RPN)

The RPN proposes object proposals along with their probability score for the Fast R-CNN to work with. It takes an image as input and outputs a set of rectangular object proposals, that is, it puts a rectangular bounding box around the detected images so that the Fast R-CNN can classify the objects in the boxes. A CNN is the first component of the RPN, and its role is to automatically extract visual features from the image by passing it through its different layers; the CNN outputs a feature map. After this, a number of Convolutional layers with different spatial window sizes slides over the feature map in order to generate object proposal; each of these spatial windows are associated with nine rectangular boxes called anchors. In introducing Faster R-CNN, [18] recommends nine anchors which are composed of three different widths and heights. Figure 4 shows the schematic architecture of the RPN. An anchor either passes as a proposed region or part of the background depending on its closeness to the ground truth. This is determined by the Intersection-over-Union (IoU), an anchor is labelled as positive if its IoU score is higher than a threshold of 0.7 or in the case of multiple ground truth, if the IoU score is the highest [18]. Boxes with IoU ration lower than 0.3 are labelled as background and the other anchors are not used for training. Given a ground truth, the IoU is the ratio of the area of overlap between the proposed region and the ground truth to the area of union; this is pictorially shown in Figure 5.

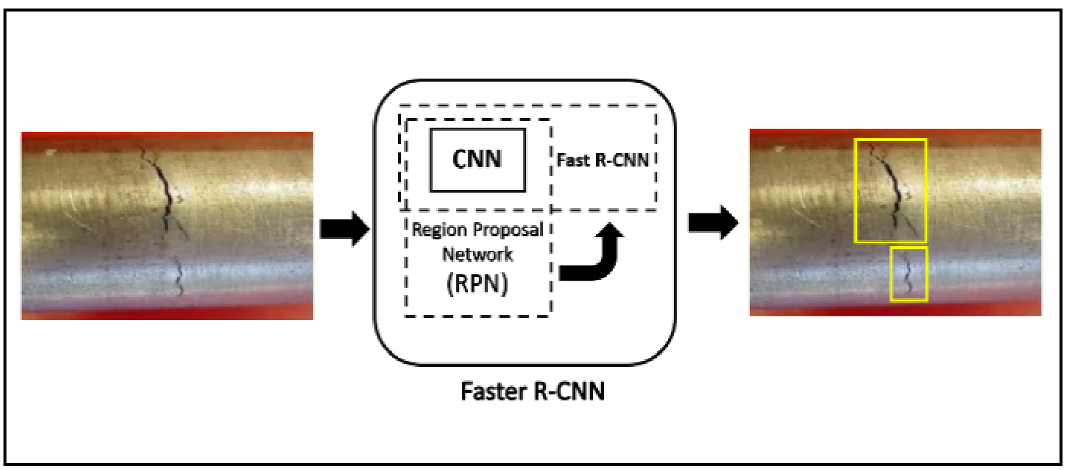

FIGURE 4: Adapted schematic architecture of the RPN. [12].

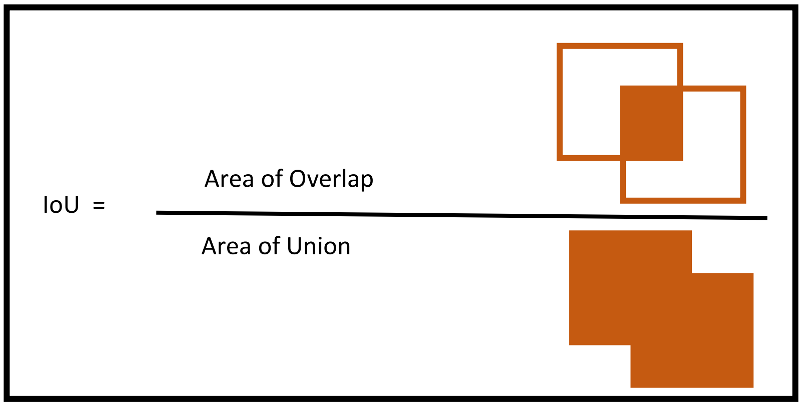

FIGURE 5: Intersection-over-Union (IoU).

The convolutional layer of different sizes is followed by a ReLU as an activation function before being fed into the Fully Connected layer, or the Feature vector. The softmax layer uses the feature vector to calculate for each of the nine generated boxes at each sliding window of the convolutional layer, the probability of being an object in the box or the probability of having no object, being part of the background; the probability is between zero and one. The regression layer, which also feeds off from the feature vector, predicts the center coordinates, width, and height of a bounding box, and is trained to map the predicted box to a ground truth box.

3.2 Fast R-CNN

The Fast R-CNN is a previous version of the Faster R-CNN proposed by [20]. Fast R-CNN first extracts visual features of the image using a CNN. Region proposals are generated using any external region proposal method, such as selective search, and are combined with the feature map from the CNN to form rectangular boxes for object detection. These Regions of Interest (RoI) are warped to a fixed size vectors through the ROI pooling layer by applying max pooling on the RoIs. These vectors are then fed to the fully connected layers followed by two regression and softmax layers for localization and classification. Figure 6 describes the architecture of a Fast R-CNN.

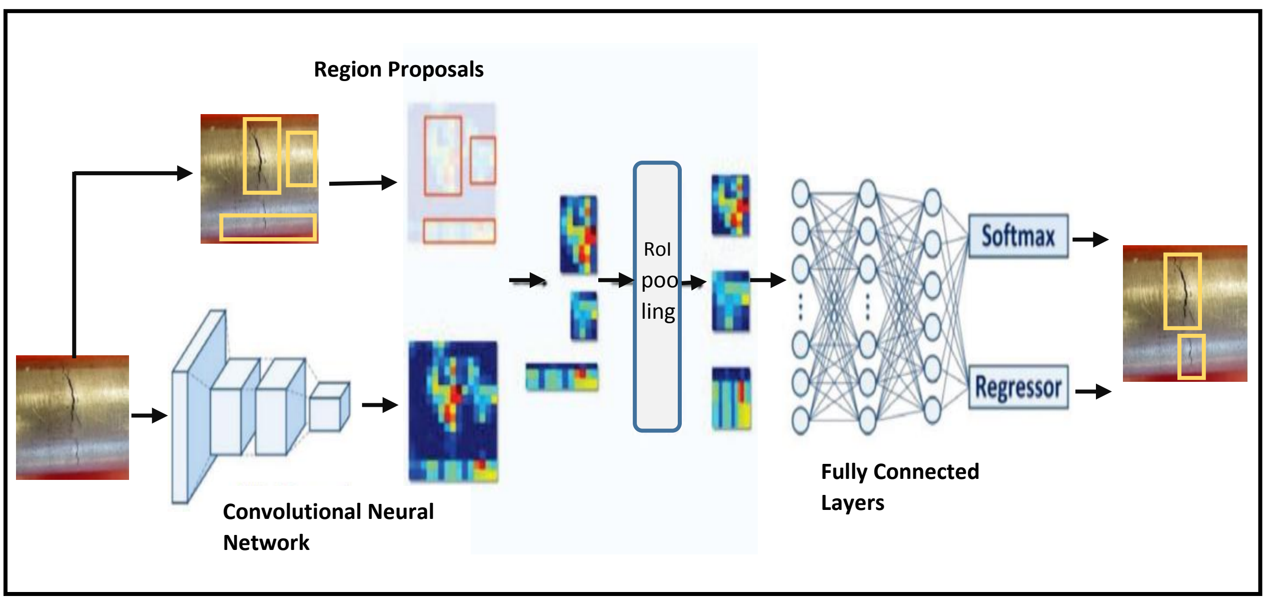

FIGURE 6: The Schematic architecture of Fast R-CNN [12].

3.3 Faster R-CNN

Having discussed the architecture of the RPN and the Fast R-CNN, how they both work together to form the Faster R-CNN shall be discussed next. In this model, computations of the CNN for feature extraction are shared by both the RPN and Fast R-CNN. The training process for this model is of four steps. The RPN is initialized with an ImageNet pre-trained model and object proposals are prepared for Fast R-CNN. In the second step, the Fast R-CNN is initialized with the trained weights of step one; in the next step, RPN is initialized with the final weights of the last step and trained again. For the last step, Fast R-CNN takes the object proposals generated in step three and is trained with the initial parameters trained in step three. In a situation where the RPN produces more than 2000 object proposals, the first 300 proposals with the highest scores are used in order to increase detection speed [18]. The Faster R-CNN is described in Figure 7. The base CNN network used is the Inception V2 [19].

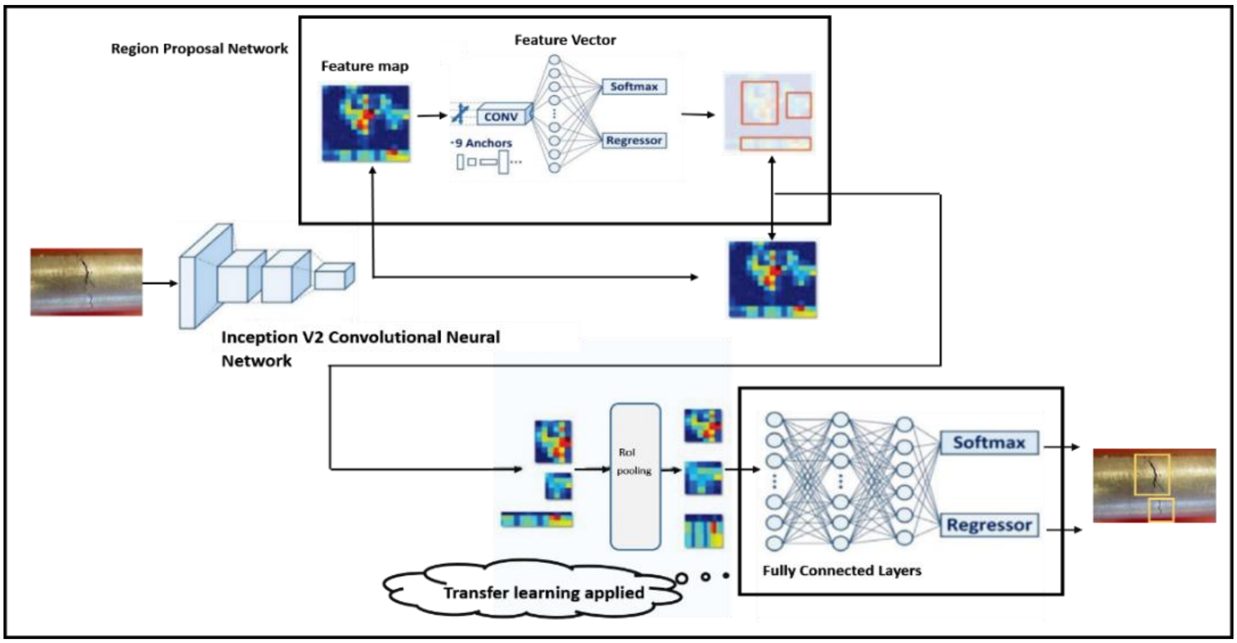

FIGURE 7: Adapted Schematic architecture of Faster R-CNN [12].

In order for this already developed Object detection model to be able to detect cracks in oil pipelines, transfer learning was applied to the model. Transfer learning makes it possible to use the weights from the layers of an already pre-trained network for a task that the network was not originally designed to do. The first step is to divide the layers into two: Early layers and Last layers. The Early layers are the layers that learn low level features from images, such as edges, colors, blobs, etc. while the Last layers learn, on a higher level, the object specific features; these are the layers that use information from the earlier layers to make out different objects. After the separation, we replace the Last layers with new layers that are able to learn the features specific to cracks in order to detect it. Following this, we combine all the layers and train the network with the training images, then we test and evaluate the network. Our transfer learning workflow is described in Figure 8.

4. DATA COLLECTION AND IMPLEMENTATION

In order to meet the objective of making a location invariant model, our dataset needs to comprise of images from different locations and under various unconstrained conditions. Our database was formed by scraping various webpages owned by oil pipeline related companies situated at different locations, for images of cracked oil pipeline. Hence, images from diverse locations such as Nigeria, Aberdeen, Canada, China, Singapore, were collected. A total of 52 images were collected and were divided in a 70%:30% proportion for training and testing. According to Figure

2, we see that the pre-processing stage has two phases, being Image augmentation and Image annotation.

4.1 Image Augmentation

This is the first phase of the Pre-processing stage. Data augmentation is a way to increase the number of data in a small dataset in order to aid the CNN to learn better by generalizing rather than overfitting, which is a danger in training a model with a small dataset. Making a CNN more robust means training it to recognize an object in different unconstrained conditions, so that if the object appears in any altered form, the model will still be able to detect it. Data augmentation entails altering and distorting the original image

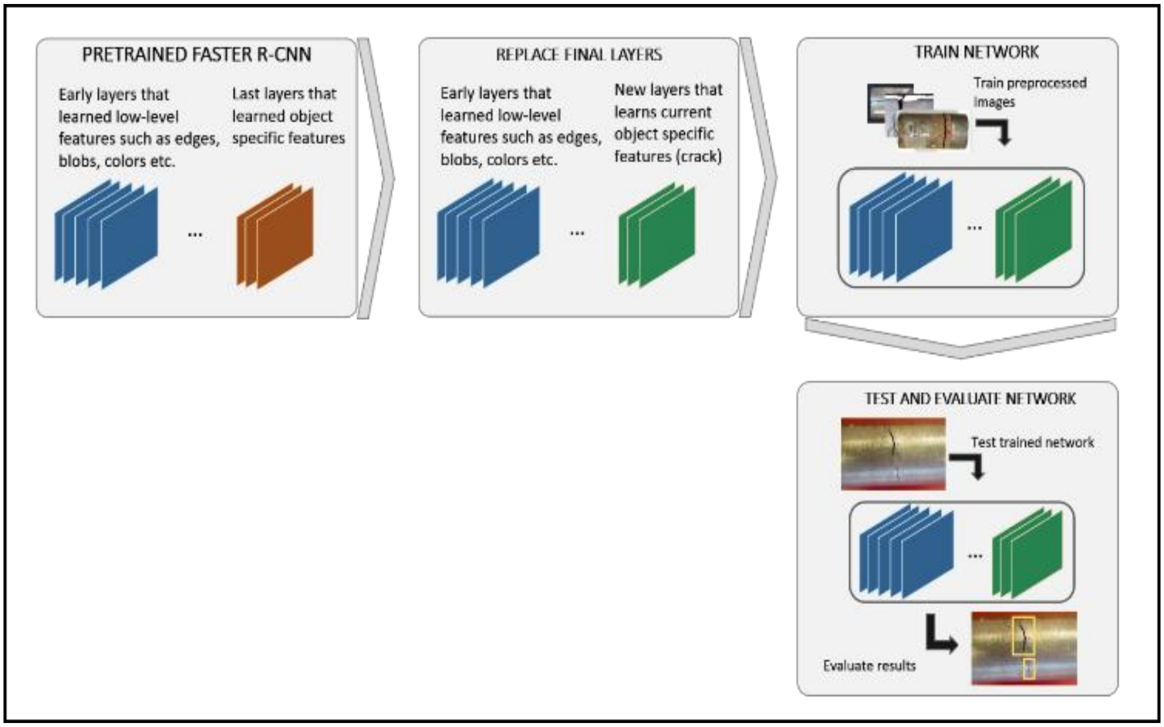

FIGURE 8: Adapted Transfer Learning workflow [21].

in a number of ways; this includes: Horizontal and vertical flipping, Scaling and Translating, Rotation and Shearing. This augmented data is used in training the model in order to enable it detect the image irrespective of any unconstrained conditions, making the model more robust. The Images were augmented using Keras, which is a neural network library written in python. Each of the images in the training dataset were augmented by generating 10 randomly altered images from them. That is, each of the images were either flipped, scaled, translated, rotated or sheared to produce about 10 augmented images which totaled 256 training images.

4.2 Image Annotation

For the crack detection and localisation model to know where an object (a crack in our case) is situated in an image, the objects in the images which will be used to train the image must first be manually labelled, by getting the coordinates bounding the object in the image; this information will then be attached to the image as a metadata when training the model. This process is also known as annotating the image. This annotation was done for all the training images (including the resulting augmented images). The coordinates are represented in an integer data format. The output of each image after annotation is of the form:

Image, w, h, object, x1, y1, x2, y2

Where w is width, h is height, and x1, y1, x2, y2 represents the coordinates of the bounding box around the object. The Image annotation stage was done using LabelImg, which is a Graphical Image annotation tool written in python. It allows for a bounding box to be drawn around the desired object by the user and attaches the coordinates of the bounding box as a metadata to the image; it outputs this as an xml file. The different xml files are collated into one csv file which is used for training.

4.3 Implementation Details

Although there are a number of programming languages available for use, the language adopted is Python. The reason Python was adopted is because of the numerous Packages, Libraries and support provided for various Machine Learning and Deep Learning tasks. The Model was developed using Tensorflow, which is a computational framework for building machine learning models; it is able to utilize the Graphic Processing Unit (GPU) of a machine to speed up large computations, making this framework suitable for Deep Learning tasks, which are involved with large computations. We are able to use this framework through its Python API. An implementation of the Faster R-CNN based on the paper by [18] was done on Tensorflow by Google, called the Tensorflow Object Detection API. This was done as an open source framework that makes it easy to construct, train and deploy object detection models. The reason this is used is because we are applying transfer learning on the already learned weights of the implemented Faster R-CNN model. The implemented model was trained on the Common Objects in Context (COCO) dataset, which is a large-scale, object detection, and captioning dataset with about 80 different object classes; this dataset doesn't contain the object we desire to detect. The layers of the Faster R-CNN were separated into Low level layers and High level layers, before replacing the high level layers with new layers and combining all the layers; the model is now ready for training.

5. TRAINING AND TESTING

The model was trained and validated with the annotated images on an Acer Nitro 5 machine with Intel(R) Core(TM) i5-8300H CPU @ 2.30GHz (8 CPUs), 8 GB DDR4 SDRAM and 4 GB NVIDIA GeForce GTX 1050 Graphics processing Unit (GPU). The hyper-parameters used in training the model are: a momentum of 0.9 and a learning rate of 0.0002. The time taken to train the model was about 4 hours; the training time was boosted because of the GPU mode used in training. For a CPU mode, it will approximately take over 4 days to train. A new set of 12 images taken from different locations were used for testing the model. Figure 9 shows the different results of some of the test images. Obviously, the model outputs a bounding box around the detected crack, including the Average Precision per test image.

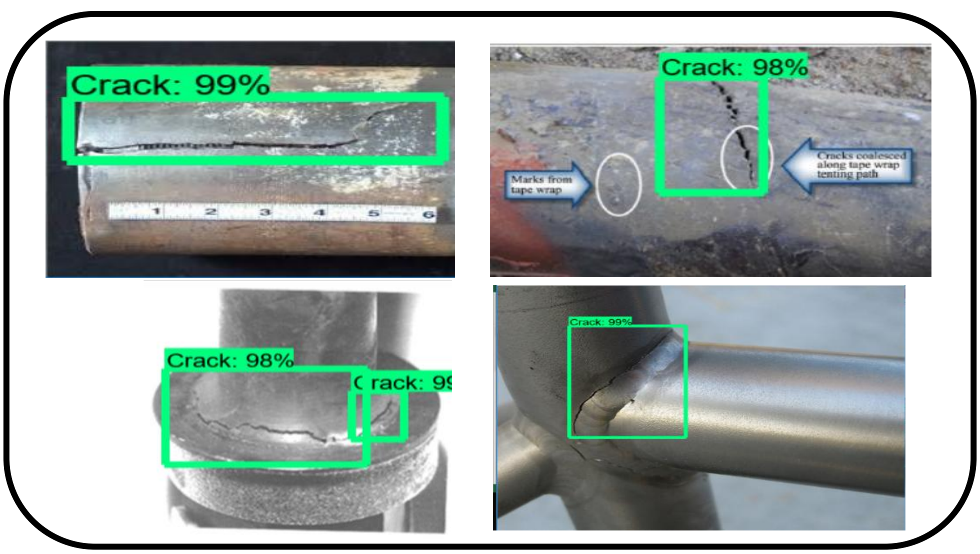

FIGURE 9: Results of the Crack detection and Localisation model.

5.1 Discussion

The graph in Figure 10 shows the Average Precision (AP) for 10 of the new test image set with the highest AP at 99% and the lowest AP at 86%. The Mean Average Precision (mAP) is 97.3%. A major challenge of the model is that it falsely detect cracks in image portions that have no cracks as shown in Figure 11. This challenge can be tackled by providing the model with more training images so it can learn to distinguish crack features from similar features which are not cracks. Out of the 12 new test images, the ratio of correctly detected images to falsely detected images is 10:2.

The Accuracy of the model based on the 12 test images is 83%, given by equation 1

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}} \qquad (1)$$

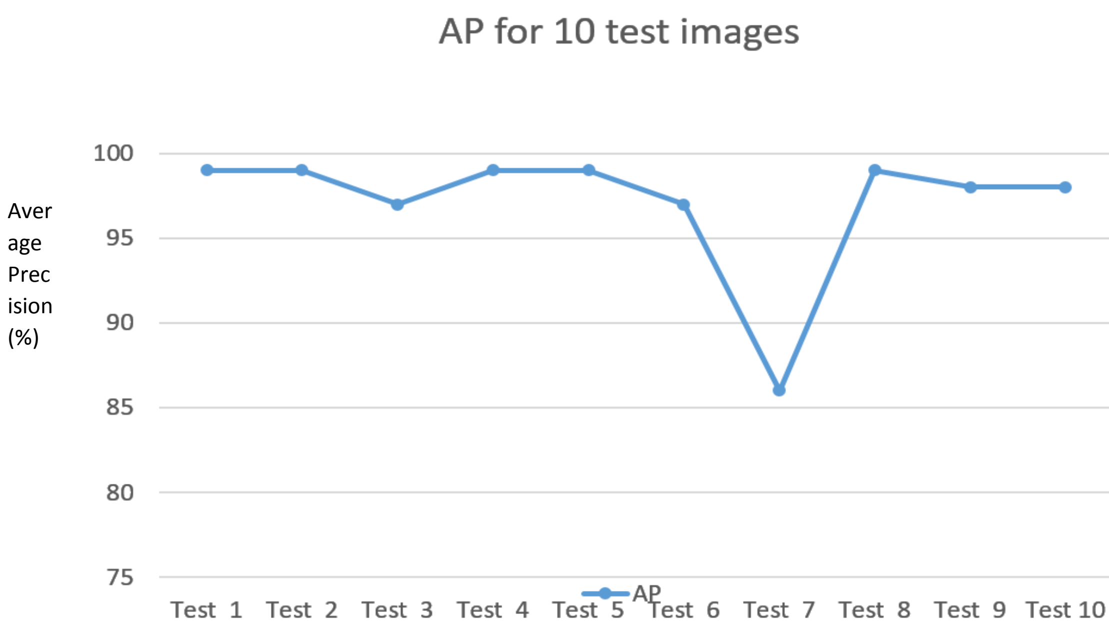

FIGURE 10: Performance of the Crack detection and Localisation Model for testing set.

5.2 Comparative Analysis

Comparing this Faster R-CNN based crack detection and localisation model with the CNN based pavement crack detection model of [14], some similarities and differences are noticed. Although the crack detection model [14] has a high accuracy, as stated in the review of existing related works, there is no indicator to aid the identification of the crack location in the image; however, the model developed in this research puts a bounding box around the crack to aid easier detection. Another point of note is that the model by [14] was reported to perform less when the testing and training images were taken from a different location than how it performed when the testing and training images were from the same location. However, the performance of the model developed in this work is not affected by test images taken from different locations.

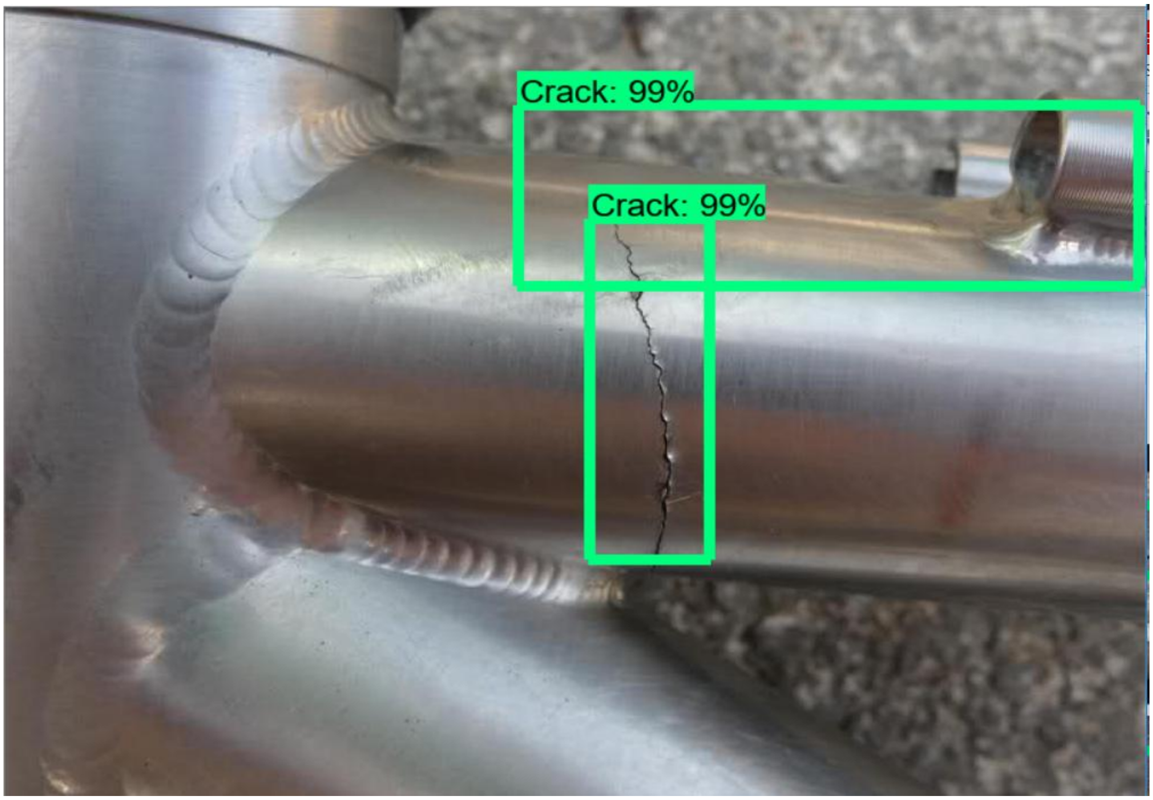

FIGURE 11: Falsely detected cracks detected the crack detection and Localisation model.

6. CONCLUSION AND RECOMMENDATION

In summary, this work focuses on the building of a model for detecting and localizing cracks in unconstrained oil pipeline images using Deep Convolutional Neural Network. The model was first designed by applying Transfer learning to a state-of –the-art CNN based object detection model - Faster R-CNN. This was implemented on Python through the Google Object detection API. Subsequently, the model was trained and tested on images gotten from various locations by scraping the web after passing them through a pre-processing stage. Finally, the model was evaluated using the mean Average Precision (mAP). The results show a bounding box around the detected crack with a mAP of 97.3% on a set of 12 new test images; the highest AP at 99% and the lowest AP at 86%. Compared to the crack detection model by [14] which adequately detects crack alone, the model developed in this work is able to localize the detected cracks, thus improving crack identification. Secondly, there is no drastic reduction in performance for test images taken from different location, making the model location invariant. However, despite its high performance, there were few cases of falsely detected cracks identified. The scope of this work was limited only to unconstrained oil pipeline images, hence, the performance of the model on other surface and structure type is unknown as of this publication; it is recommended for further research. The experimental results of this work shows its potential in curbing the dangerous effects of oil spillage in the oil and gas sector by detecting cracks in oil pipelines. If adopted, this research can potentially aid in eliminating the drawbacks associated with current crack detection systems that are highly dependent on human input by reducing cost and man power (among others) while crack inspections are carried out.

Further work on this research could address the false detection problem, in order to enhance the model's detection accuracy. Also, the model could be improved for detection of more defects of oil pipeline such as, corrosion, punctures and dents. The model could also be trained to detect cracks in multiple surfaces and structures.

7. REFERENCES

[1] T. A. Abii & P. C. Nwosu. "The Effect of Oil-Spillage on the Soil of Elemets in Rivers State of the Niger-Delta Area of Nigeria." *Research Journal of Environmental Sciences*, vol. *3*(3), pp. 316–320. 2009.

[2] O. E. Inoni, D. G. Omotor & F. N. Adun. "The Effect of Oil Spillage on Crop Yield and Farm Income in Delta State, Nigeria." *Journal of Central European Agriculture*, vol. *7*(1), pp. 41–48, 2006.

[3] A. A. Kadafa. "Oil Exploration and Spillage in the Niger Delta of Nigeria." *Civil and Environmental Research*, vol. *2*(3), pp. 38–51. 2012.

[4] T. Yamaguchi & S. Hashimoto. "Automated crack detection for concrete surface image using percolation model and edge information." *IECON Proceedings (Industrial Electronics Conference)*, 2006, pp. 3355–3360.

[5] E. G. Hinton, & S. Osindero. "A Fast Learning Algorithm for Deep Belief Nets." *Neural Computation,* vol. 18(7), pp. 1527-1554, 2006.

[6] N. Tajbakhsh, J. Y. Shin, S. R. Gurudu, R. T. Hurst, C. B. Kendall, M. B. Gotway, S. Member. "Convolutional Neural Networks for Medical Image Analysis : Full Training or Fine Tuning ?" *IEEE Transactions on Medical Imaging*, vol. *35*(5), pp. 1299–1312, 2016.

[7] T. G. Davies & M. S. Mamlouk. "Theoretical response of multilayer pavement system to dynamic non-destructive testing." *Transportation Research Record*, vol. 1022, pp. 1–7, 1985

[8] S. K. Sinha, P. W. Fieguth & M. A. Polak. "Computer Vision Techniques for Automatic Structural Assessment of Underground Pipes." *Computer-Aided Civil and Infrastructure Engineering,* vol. 18(2), pp. 95-112, 2003.

[9] S. Iyer & S. K. Sinha. "Segmentation of Pipe Images for Crack Detection in Buried Sewers." *Computer-Aided Civil and Infrastructure Engineering*, vol. 21(6), pp. 395–410, 2006.

[10] M. Salman, S. Mathavan, K. Kamal & M. Rahman. "Pavement crack detection using the Gabor filter." *16th International IEEE Conference on Intelligent Transportation Systems*, 2013, pp. 2039–2044.

[11] Y. Noh, D. Koo, Y. Kang, D. Park & D. Lee. "Automatic Crack Detection on Concrete Images Using Segmentation via Fuzzy C-means Clustering." *2017 IEEE International Conference on Applied System Innovation.* 2017, pp. 877–880.

[12] Y. Cha, W. Choi, G. Suh & S. Mahmoudkhani. "Autonomous Structural Visual Inspection Using Region-Based Deep Learning for Detecting Multiple Damage Types." *Computer-Aided Civil and Infrastructure Engineering*, vol. *33*(9), pp. 731-747, 2017.

[13] A. Zhang, K. C. P. Wang, B. Li, E. Yang, X. Dai, Y. Peng, C. Chen. "Automated Pixel-Level Pavement Crack Detection on 3D Asphalt Surfaces Using a Deep-Learning Network." *Computer-Aided Civil and Infrastructure Engineering*, vol. *32*(10), pp. 805–819, 2017.

[14] L. Pauly, H. Peel, S. Luo, D. Hogg & R. Fuentes. "Deeper Networks for Pavement Crack Detection." *Proceedings of the 34th International Symposium on Automation and Robotics in Construction (ISARC*), 2017, pp. 479-485.

[15] K. Gopalakrishnan, S. K. Khaitan, A. Choudhary & A. Agrawal. "Deep Convolutional Neural Networks with transfer learning for computer vision-based data-driven pavement distress detection." *Construction and Building Materials,* vol. 157, pp. 322–330, 2017.

[16] H. Maeda, Y. Sekimoto, T. Seto, T. Kashiyama & H. Omata. "Road Damage Detection Using

Deep Neural Networks with Images Captured Through a Smartphone." *Computer-Aided Civil and Infrastructure Engineering*, vol. 33(12), pp. 1127-1141, 2018.

[17] H. Goh, N. Thome, M. Cord & J. Lim. "Learning Deep Hierarchical Visual Feature Coding." *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25(12), pp. 2212-2225, 2014.

[18] S. Ren, K. He & R. Girshick. "Faster R-CNN : Towards Real-Time Object Detection with Region Proposal Networks*." IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39(6), pp. 1137-1149, 2016.

[19] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens & Z. Wojna . "Rethinking the Inception Architecture for Computer Vision." *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR*), 2016, pp. 2818-2826.

[20] R. Girshick. "Fast R-CNN." *2015 IEEE International Conference on Computer Vision (ICCV*). 2014, pp. 1440–1448.

[21] Mathworks. "Transfer Learning." Internet: https://www.mathworks.com/discovery/transfer-learning.html, n.d. [Feb. 22, 2019].