Deep Learning for Underwater Pipeline Corrosion Detection: A Comparative Analysis of CNN Architectures

[1]Dr. Anitha T N, [2]A Abhilash,[3]Adarsh C R,[4]Aishwarya D, [5]Megha

[1]HOD & Professor, [2]Student, [3]Student, [4]Student, [5]Student
[1]Department of Computer Science & Engineering,
[1]Sir M Visvesvaraya Institute of Technology, Bangalore, India

***Abstract*: In this study report, four different Convolutional Neural Network (CNN) algorithms are used to investigate underwater pipeline corrosion detection. As corrosion poses a growing threat to underwater infrastructure, early detection is essential to avert expensive losses and environmental risks. This work intends to investigate CNNs' effectiveness in identifying corrosion in underwater pipes by utilizing their capabilities, which have demonstrated promise in image recognition tasks. The backdrop of underwater pipeline corrosion, the significance of early detection, and CNN algorithms as a potential remedy are all covered in this research. A thorough analysis of the body of research on CNN applications and corrosion detection techniques in related fields is offered. The dataset utilized for testing and training, as well as the particulars of the CNN algorithms used, are described in the methods section. Experimental data and discussions, including comparisons of accuracy, precision, recall, and F1 score, show how well each CNN algorithm performs. By contrasting the suggested CNN-based method of underwater corrosion detection with more recognized methodologies, the study effectively highlights the advantages and disadvantages of the technology. The conclusion includes a synopsis of the key findings of the study, recommendations for other research directions, and implications for underwater pipeline maintenance.**

Numerous businesses, including oil and gas, telecommunications, and renewable energy, depend heavily on underwater pipelines. But corrosion also endangers a ship's strength, much as weather and time can erode a ship's hull, leading to leaks, damaging the environment, and necessitating costly repairs. Usually, manual examination is required, which is costly, time-consuming, and prone to errors. Consequently, there is an increasing need for trustworthy, economical, and effective methods of automatically identifying corrosion. In recent years, Convolutional Neural Networks (CNNs) have shown to be a very successful technology for image recognition applications.

They are perfect for pattern detection in photographs, including underwater camera photos, because of their special capacity to automatically generate hierarchical representations from raw data. The potential of CNNs to address the issues related to underwater pipeline corrosion detection is examined in this study.

The study starts with a comprehensive analysis of the body of research on corrosion detection techniques and CNN applications in related domains. Understanding the state-of-the-art now and spotting gaps in the literature that this study seeks to fill are made easier with the help of this review. Building on this understanding, the methodology section explains the four CNN algorithms that were chosen for evaluation, their architectural characteristics, and the dataset that was utilized for training and testing the CNN models.

The test results demonstrate how well the CNN-based approach detects damage to underwater pipelines. Each CNN algorithm is assessed using a variety of performance metrics, providing helpful information about its benefits and drawbacks. The benefits of CNNs over antiquated corrosion detection methods are also discussed, emphasizing how quickly and precisely they can assess massive volumes of image data.

In summary, our research contributes to ongoing efforts to enhance the maintenance and observation of underwater pipelines. It offers a useful technique for early corrosion identification using CNNs, helping to protect critical infrastructure and the environment. Further research directions could include improving the CNN architecture, looking into new features to improve detection accuracy, and putting the recommended approach into practice and confirming it in real-world situations.

I. INTRODUCTION

A costly problem that drains companies' resources and gives them issues is corrosion. It might be compared to the gradual deterioration of materials because of chemical interactions with their surroundings. Underwater pipelines are vital conduits for gas, oil, water, and even the signals that keep us linked, and they are among the worst degraded.

Corrosion affects not only the structural integrity of these pipes but also the environment and economics. We explore the complicated realm of underwater pipeline corrosion in this introduction, illuminating its causes, consequences, and the pressing need for improved detection and monitoring techniques.

Underwater pipelines are essential routes for the movement of water, gas, oil, and communications signals. They are a fundamental part of many global economies and industries. But corrosion poses a constant threat to these underwater structures. Underwater pipeline integrity, safety, and longevity are seriously threatened by corrosion, the slow degradation of materials brought on by chemical or electrochemical reactions with the environment. We examine the origins, symptoms, and extensive ramifications of underwater pipeline corrosion in this introduction.

Because of their high salinity, fluctuating temperatures, and the availability of corrosive chemicals such dissolved oxygen and chloride ions, underwater settings offer a very corrosive environment. Particularly seawater acts as a strong corrosion catalyst, hastening the deterioration of metallic materials—like steel—that are frequently utilized in pipeline construction. The presence of both oxygen and chloride ions in seawater encourages the production of electrolytes that are corrosive, which can cause pitting, cracking, and eventually structural failure. Furthermore, the presence of microorganisms in marine environments, such as biofilm formation and the occurrence of microbiologically influenced corrosion (MIC), intensifies corrosion rates, and makes detection more difficult.

Numerous detrimental repercussions of underwater pipeline corrosion are seen in society, the economy, and the environment. Corrosion causes structural degradation that compromises pipeline integrity and raises the possibility of leaks, spills, and ruptures. Because they upend coastal communities, endanger aquatic life, and harm maritime environments, these catastrophes pose serious environmental risks. Pipeline malfunctions can result in significant replacement and repair costs, lost revenue from operations being halted, and occasionally even legal ramifications. Pipeline failures affect public safety because mishaps like oil spills can seriously harm people's health and well-being.

Figure 1. Corroded Pipeline
Courtesy: google

Underwater pipeline corrosion presents several issues that call for a multimodal strategy that includes mitigation, detection, and prevention techniques. Although coatings, cathodic protection, and material selection are important corrosion control strategies, early detection is just as important for seeing and treating corrosion before it worsens and causes catastrophic failures. Particularly in underwater situations, the accuracy, coverage, and efficiency of conventional corrosion detection techniques, such as visual inspections and physical tests, are constrained.

Promising paths to improve underwater pipeline corrosion detection and monitoring are provided by emerging technologies. Using remote sensing tools, such as underwater cameras and acoustic sensors, high-resolution data from submerged pipes may be collected, which makes it easier to spot corrosion-related anomalies early on. Convolutional Neural Networks (CNNs) are a breakthrough in machine learning and data analytics that present prospects for automated corrosion classification and detection based on picture data.

In this regard, the purpose of this study is to investigate how well CNN algorithms work for identifying and evaluating corrosion in underwater pipes. We want to create robust and economical corrosion detection models that may supplement current monitoring tactics and help safeguard underwater infrastructure and marine habitats by utilizing CNNs' capacity to handle and understand enormous volumes of underwater video. We aim to further the understanding of underwater pipeline corrosion and contribute to the development of creative solutions for its detection and mitigation through empirical assessments and comparative analyses.

II. EXISTING SYSTEM

2.1 Ground Penetrating Radar

To penetrate the surface of materials utilized for corrosion detection, ground penetrating radar (GPR) uses electromagnetic reflection. Within the frequency range of 500 MHz to 2.5 GHz, a brief, tiny electromagnetic energy pulse is transferred to the surface. GPR antennae's reflected waves are used to record the amplitude-time signal. GPR has several benefits, including as quick scanning, sensitivity to materials embedded in the structure, high-quality pictures of the structure, and deeper penetration. But because GPR data can be quite challenging to comprehend, they need to be handled by operators with specialized skills. It also needs to be done to capture photographs with deep post-processing. Furthermore, corrosion progression cannot be tracked online using this method; it needs to be applied on a frequent basis.

GPR data, artificial neural networks, and image processing were used by researchers in to determine the level of corrosion in reinforced steel concrete structures. To look at variations, edge detection and k-means clustering image processing techniques were applied in between characteristics of rusting. In order to decipher and extract complex feature data from GPR images, ANN was utilized. In order to lessen the blurring effect on GPR photographs, the photos were shot quickly. The outcomes shown that a corroded image could be extracted and divided into several segments using k-means clustering. The area was divided into four groups based on the semantic information that was obtained: no corrosion, low corrosion, moderate corrosion, and high corrosion. A Sobel edge detector was used to accurately detect the edges of several segment sections. Furthermore, GPR was applied with Fourier transform and a Gaussian filter with 1 GHz frequency to monitor and predict the degree of corrosion. Concrete walls, floors, and roofs that were reinforced with steel were scanned, and the resulting images were pre-processed to identify anomalies and noise. Simulators were used to validate the findings and tests using field data. This method is better at recognizing and detecting corrosion since it produced the most accurate estimate regarding the extent of rebar corrosion. Additionally, GPR can be used in conjunction with other imaging methods to increase the precision of corrosion detection. In acoustic scanning and GPR were used to identify deterioration in concrete bridge decks that were both straight and curved. The energy map was utilized to rebuild 2D images over a frequency range of 0.5 to 5 KHz. The findings demonstrated that thorough corrosion identification in the bridge decks can be achieved by integrating deterioration and acoustic scanning maps.

2.2 Color Space Detection

The color representation of an image that mathematical models describe is called color space. The properties of color spaces are used to define the image pixels. Using the color space segmentation technique, the clusters of color features in image pixels match to the significant objects in the image. The color characteristics of these image clusters are comparable. There isn't a single-color space, though, that works with every type of image. As a result, a great deal of research has been done to determine which color space works best in various settings and circumstances. RGB stands for red, green, and blue; YUV for brightness and chroma; CMY for cyan, magenta, and yellow; HSV for hue, saturation, and value; and XYZ for x, y, and z coordinates are some of the most well-known color spaces.

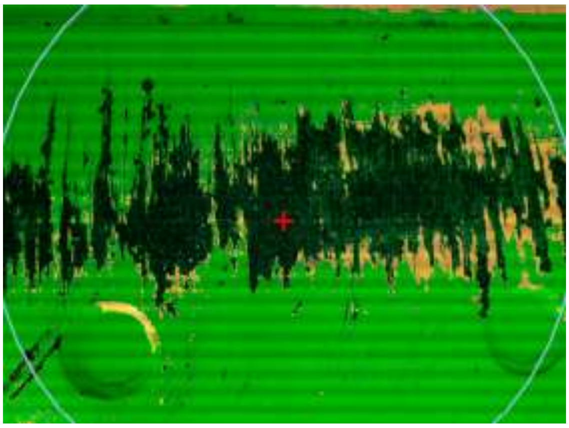

Figure 2. Corrosion detection using a RGB colors

Based on the hue and roughness of the materials, the HSV color space can be utilized to identify areas that are corroded and those that are not. In this work, a corrosion-representative histogram's color was analyzed using HSV. Additionally, roughness analysis was conducted using the grayscale level. After that, this image processing method was applied to the photographic images of the parts and structures. To increase the corrosion damage assessment's detection accuracy, artificial intelligence algorithms were integrated with this color space. Hue, saturation, and intensity, or HSI, can also be used to identify rust. Using histogram equalization, an image enhancement approach was used to the corroded areas to boost contrast and brightness. The RGB photos were used to convert the HSI. Nevertheless, by removing the CbCr (blue and red chroma) elements from the corrosion images, robust corrosion detection can be created. It is possible to differentiate between the textures of corrosion, which can help with computation and measurement of non-uniform corrosion. To secure electric energy transmission by promptly identifying the damper rust fault, a unique color space based on the color shade index (CSI) and rusty area ratio (RAR) was introduced. Grayscale pictures of the complicated dampers were obtained and then processed using local difference processing, edge strength mapping, and the Gaussian kernel. The photos were then divided into four categories: moderate rust, medium rust, severe rust, and rust-free. The findings demonstrated efficient segmentation and 93% accuracy in identifying corroded dampers.

2.3 Texture Analysis

This technique is also used in computer vision for computer image analysis's object classification. Using this method, the edge boundaries of different textured sections that have corroded in the photos are calculated. Texture analysis is a useful tool for improving classification accuracy since it minimizes errors in the detection of isolated data. The corroded materials' gradual development in surface roughness is a basic feature of this method. The surface pit depths, which give shadow lengths from a single incident light source, can be used to measure it. This thus makes it possible to determine the average corrosion rate by rebuilding a pseudo-30 material exterior. The image's edge pixels are separated from the corroded arms.

Figure 3. Texture analysis of a ship structure

These edge pixels have varying grayscale values and are located on a corroded region boundary. To accurately define a corroded boundary, the difference between the edge pixels and neighboring pixels is determined. To quantify the discrepancies between the image's grayscale values, utilize the intensity characteristic. The literature has examples of the texture analysis technique. Corrosion areas in damages can be precisely identified, detected, and categorized using the texture analysis technique. It is feasible to distinguish between areas that have corroded and those that have not. Moreover, any segmentation technique now in use, such as region-based, thresholding-based, edge detection, clustering, etc., can be combined with the texture analysis methodology. Therefore, we use the active contour segmentation approach to propose a mix of thresholding-based and edge detection methods. It has been demonstrated that active contour is accurate in segmenting objects with weak, irregular borders. This approach will therefore be adjusted by creating an initial contour, curve restrainer, and halting function to account for weak corrosion boundaries. Next, the adjusted active contour will be led to segment the accurate corrosion zones using the thresholding of pixel property approach. Figure S shows the initial outcomes of the suggested technique. This figure illustrates how corrosion limits on ship structure photos can be identified using the active contour. However, additional research on segmentation and recognition accuracy is required to evaluate the efficacy and accuracy of the suggested approach.

2.4 Ultrasound Testing

A non-destructive testing method called ultrasonic testing (UT) is used to evaluate the integrity of materials and find flaws like corrosion in pipes. It is based on sending high-frequency sound waves into the substance that is being examined. These waves pass through the material and bounce back to the surface when they come across a boundary or defect. Inspectors can find flaws, gauge the amount of corrosion, and gauge the thickness of the material by measuring the time it takes for sound waves to travel and the amplitude of the reflected signals. Because ultrasonic testing may provide accurate and reliable measurements without needing direct contact with the pipeline surface, it is particularly helpful in the context of underwater pipelines.

Inspectors can assess the state of the pipeline remotely by transmitting and receiving sound waves using specialized ultrasonic probes. Ultrasonic testing underwater is achievable with remotely operated vehicles (ROVs) equipped with ultrasonic sensors, allowing inspections to be conducted in challenging underwater environments. Because ultrasonic testing provides detailed information about the thickness and integrity of pipeline walls, it is crucial for identifying corrosion-related problems and directing maintenance and repair decisions. This contributes to ensuring underwater pipes operate safely and effectively.

III. PROPOSED SYSTEM

Using a labeled dataset as a basis, the algorithm in supervised learning learns a mapping from input data to output labels. The algorithm is given a training dataset of input-output pairs, where each input is linked to a corresponding output label, in supervised learning. Learning a function that can precisely predict the output labels given fresh, unknown input data is the aim of supervised learning.

The process of supervised learning typically involves the following steps:

1. **Data Collection:** The first step in supervised learning is to collect a labeled dataset containing input-output pairs. The input data can be in various forms, such as images, text, or numerical features, while the output labels represent the target values or classes to be predicted.

2. **Data Preprocessing:** Once the dataset is collected, it is often necessary to preprocess the data to prepare it for training. This may involve tasks such as normalization, feature scaling, or handling missing values to ensure that the input data is suitable for the learning algorithm.

3. **Model Selection:** In supervised learning, the choice of model or algorithm is crucial for achieving good performance. There are various supervised learning algorithms available, ranging from simple linear models to complex neural networks. The selection of the model depends on factors such as the nature of the data, the complexity of the problem, and the desired performance metrics.

4. **Training:** During the training phase, the algorithm learns to map the input data to the corresponding output labels by adjusting its parameters based on the training dataset. This involves feeding the training data into the model and updating its parameters iteratively using optimization techniques such as gradient descent or stochastic gradient descent.

5. **Evaluation:** After training the model, it is evaluated using a separate validation or test dataset to assess its performance. Various performance metrics can be used to evaluate the model's accuracy, such as accuracy, precision, recall, F1 score, or mean squared error, depending on the nature of the problem.

6. **Deployment:** Once the model has been trained and evaluated, it can be deployed to make predictions on new, unseen data. The trained model takes the input data as input and produces predictions or output labels based on the learned mapping from the training phase.

Convolutional Neural Networks (CNNs) are a class of deep learning algorithms specifically designed for processing structured grid-like data, such as images. They have revolutionized the field of computer vision and are widely used in various applications, including image recognition, object detection, and segmentation. CNNs are inspired by the organization of the animal visual cortex and consist of multiple layers of interconnected neurons that perform hierarchical feature extraction.

1. Convolutional Layers:
- The core building blocks of CNNs are convolutional layers, which apply filters or kernels to input images to extract features. Each filter detects specific patterns or features, such as edges, textures, or shapes, by performing convolutions across the input image. Convolutional layers use shared weights and biases, enabling them to detect the same features across different regions of the image.

2. Pooling Layers:
- Pooling layers are used to down sample the feature maps generated by convolutional layers, reducing their spatial dimensions while retaining the most important information. Common pooling operations include max pooling, which selects the maximum value within a window, and average pooling, which computes the average value. Pooling helps to make the CNN more robust to variations in input data and reduces the computational complexity of subsequent layers.

3. Activation Functions:
- Activation functions introduce non-linearities into the network, enabling it to learn complex patterns and relationships in the data. Common activation functions used in CNNs include the Rectified Linear Unit (ReLU), which replaces negative values with zero, and the Sigmoid and Hyperbolic Tangent (Tanh) functions, which squash the output to a range between 0 and 1 or -1 and 1, respectively.

4. Fully Connected Layers:
- Fully connected layers, also known as dense layers, are typically added to the end of the CNN architecture to perform classification or regression tasks. These layers connect every neuron in one layer to every neuron in the next layer, allowing the network to learn high-level representations of the input data. Fully connected layers are often followed by softmax activation functions for classification tasks or linear activation functions for regression tasks.

5. Training:

- CNNs are trained using backpropagation and stochastic gradient descent algorithms to minimize a loss function that measures the difference between the predicted and actual outputs. During training, the network learns to adjust its weights and biases iteratively by propagating errors backward through the layers. Training data is typically divided into batches to accelerate the optimization process and reduce memory requirements.

CNNs have demonstrated remarkable performance in various computer vision tasks, achieving state-of-the-art results in image classification competitions and real-world applications. Their ability to automatically learn hierarchical representations of visual data makes them invaluable tools for analyzing and understanding images in diverse domains. Convolutional neural networks (CNNs) and deep learning have made significant strides in computer vision research in recent years. We can now receive and interpret visual input for tasks like object detection, semantic segmentation, and image categorization with never-before-seen levels of efficiency and accuracy thanks to these sophisticated models. Among all the CNN designs created, VGG16, ResNet, MobileNet, and DenseNet stand out as important trailblazers that have advanced deep learning and brought new insights to the industry.

3.1.VGG16 (Visual Geometry Group 16): The VGG16 architecture is a convolutional neural network designed by the Visual Geometry Group at the University of Oxford University. What makes it unique is its deep architecture, which is made up of 16 weight layers (three fully linked layers and thirteen convolutional layers). VGG16 is known for its simplicity and uniformity since it uses tiny (3x3) convolutional filters across the network along with max-pooling layers.

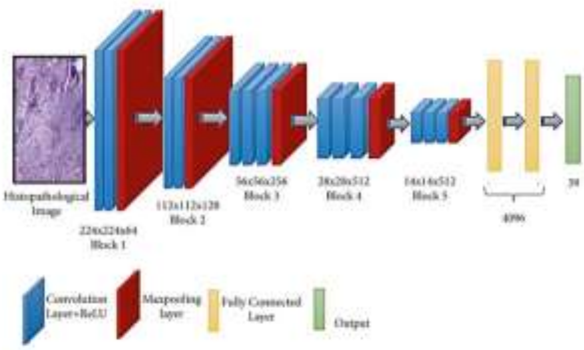

Figure 4. VGG16 architecture Google Curtesy

Max-pooling layers reduce spatial dimensions sequentially after the several convolutional layers with increasing depth in the VGG16 architecture. Iterating this pattern several times results in fully connected layers at the end for classification. Rich hierarchical representations of visual input may be extracted using VGG16, which is why many computer vision applications choose it for feature extraction.

3.2.ResNet (Residual Network): ResNet is a revolutionary convolutional neural network architecture proposed by Microsoft Research. The concept of residual learning was introduced, whereby shortcut connections—also referred to as skip connections—are added to the network to allow information to flow directly from earlier levels to subsequent layers. In doing so, the problem of the vanishing gradient is solved, enabling the training of extraordinarily deep networks—hundreds of layers—while maintaining efficiency.

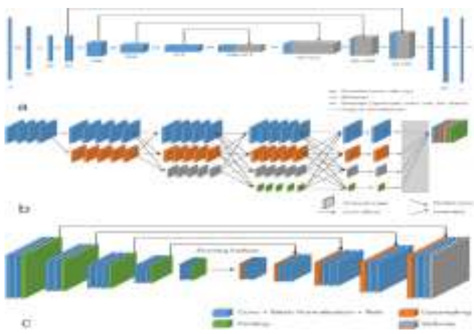

Figure 5.Resnet architecture
Courtesy: google

The primary innovation of ResNet is the use of residual blocks, which consist of many convolutional layers followed by a shortcut link that adds the original input to the output of the convolutional layers. This residual learning framework allows the network to learn residual functions, enabling more in-depth training and optimization of the network. ResNet architectures—like ResNet50 and ResNet101—have grown in popularity.

3.3.MobileNet: A convolutional neural network architecture called MobileNet is lightweight and intended for embedded and mobile devices with constrained processing power. It is notable for using depth wise separable convolutions to achieve both efficiency and compactness; it was first described by Google Research.

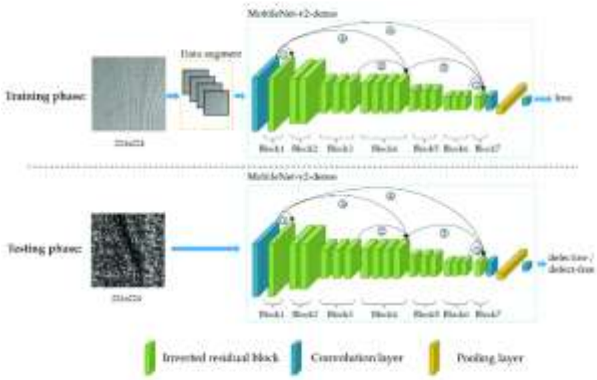

Figure 6.Mobilenet architecture Google Curtesy

Depth wise separable convolutions divide ordinary convolutions into two distinct layers: depth wise convolutions that individually apply a single filter to each input channel and pointwise convolutions that use 1x1 convolutions to aggregate the outputs of the depth wise convolutions. Thus, convolutions use less processing power while maintaining representational capability. Real-time systems and mobile apps have embraced MobileNet designs, like MobileNetV1 and MobileNetV2, for tasks like semantic segmentation, object detection, and photo categorization.

3.4.DenseNet (Densely Connected Convolutional Networks): Researchers at Cornell University have created a convolutional neural network design called DenseNet. It is distinguished by a dense connectivity structure in which all layers are feed-forwardly coupled to all other layers. Because of its extensive connection, very deep networks can be trained more effectively because it promotes gradient flow and feature reuse across the network.

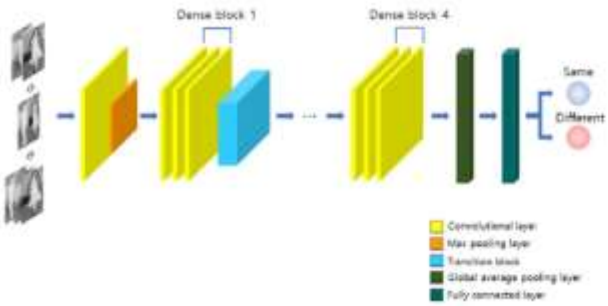

Figure 7.Densenet architecture
Courtesy: google

Each layer in DenseNet concatenates feature maps along the channel dimension after receiving them as input from all previous layers. Reusing features and increasing representational capacity are made possible by this extensive connectedness, which guarantees that every layer has access to every feature map that has been learnt by layers before it. DenseNet architectures have been widely used in academic and industry applications. Examples of these architectures are DenseNet-121 and DenseNet-169, which have shown state-of-the-art performance on a variety of image classification benchmarks.

IV. RESULTS

4.1.VGG16

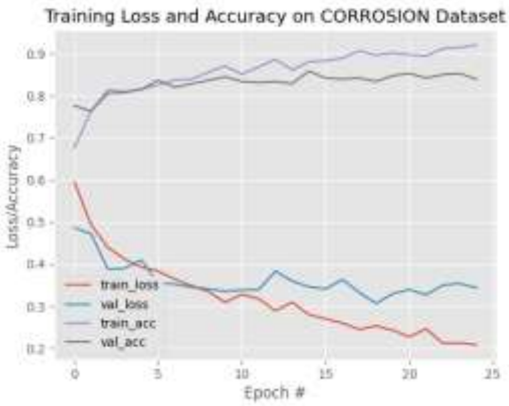

Figure 8. Line graph depicting results of VGG16 model.

The line graph depicted above illustrates a VGG16 model's accuracy and training loss using a corrosion dataset.
The number of times the training data has been run through the model is displayed on the "Epoch #" labeled x-axis of the graph. The graph's y-axis is titled "Loss/Accuracy" and has two lines on it: train_acc (training accuracy) and train_loss (training loss). On the same graph, lines representing validation accuracy (val_acc) and validation loss (val_loss) are also displayed.
As more training epochs are added to the model, the training loss and accuracy curves usually exhibit a declining trend. This indicates that the model is improving its ability to match the training set. The validation loss and accuracy curves are employed to track the model's ability to generalize to previously unseen data. Although the validation loss and accuracy curves should ideally also have a downward trend, if the model is beginning to overfit the training data, they may begin to plateau or even slightly increase.
It appears from the image you gave that the model is operating effectively on the corrosion dataset. Training accuracy is rising while training loss is falling. Along with the validation accuracy rising, there is a decrease in validation loss. This suggests that the model is generalizing well to unseen data.
The following more information can be deduced from the picture:
Compared to the validation accuracy, the training accuracy is higher. This is normal as, by definition, the model can always fit the training set exactly. It is important to note that a significant discrepancy between the accuracy of training and validation should not be present since this may suggest overfitting of the model.
The range of the y-axis is 0 to 1. This shows that the accuracy is calculated as the percentage of correct classifications, while the loss is measured using a binary cross-entropy loss function.
Overall, based on the image you gave, it appears that this dataset would work well with the VGG16 model for corrosion detection.

4.2.MobileNet

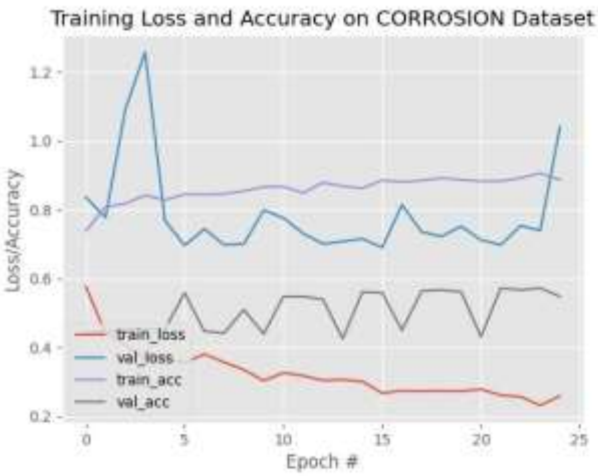

Figure 9. Line graph depicting results of MobileNet model

The figure above shows a line graph that shows the accuracy and training loss of a MobileNet model using a corrosion dataset.

On the graph's "Epoch #" labeled x-axis, the number of times the training data has been run through the model is shown. The two lines on the "Loss/Accuracy" y-axis of the graph are labeled "train_acc" for training accuracy and "train_loss" for training loss. Lines showing validation loss (val_loss) and accuracy (val_acc) are also shown on the same graph. As more training epochs are added to the model, the training loss and accuracy curves usually exhibit a declining trend. This indicates that the model is improving its ability to match the training set. The validation loss and accuracy curves are employed to track the model's ability to generalize to previously unseen data. Although the validation loss and accuracy curves should ideally also have a downward trend, if the model is beginning to overfit the training data, they may begin to plateau or even slightly increase.
It appears from the image you gave that the model is operating effectively on the corrosion dataset. Training accuracy is rising while training loss is falling. Along with the validation accuracy rising, there is a decrease in validation loss. This suggests that the model is generalizing well to unseen data.
The following more information can be deduced from the picture:
Compared to the validation accuracy, the training accuracy is higher. This is normal as, by definition, the model can always fit the training set exactly. It is important to note that a significant discrepancy between the accuracy of training and validation should not be present since this may suggest overfitting of the model.
The range of the y-axis is 0 to 1. This shows that the accuracy is calculated as the percentage of correct classifications, while the loss is measured using a binary cross-entropy loss function.
Overall, based on the image you gave, it appears that this dataset lends itself well to the MobileNet model for rust identification. A lightweight deep learning model called MobileNet is ideal for embedded systems or mobile applications. This makes it a viable option for real-time corrosion detection, where it could be preferable to use a model on autonomous cars or underwater sensors.

4.3.Densenet

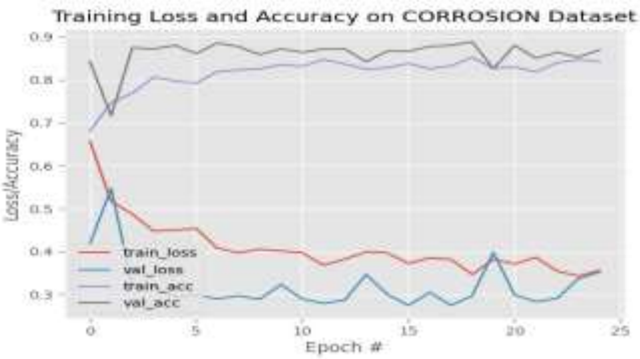

Figure 10. Line graph depicting results of DenseNet.

The picture mentioned above is a line graph that illustrates a DenseNet model's accuracy and training loss using a corrosion dataset. Although the text label on the y-axis is partly obscured, it most likely reads "Loss/Accuracy" according to standard corrosion detection graphs.
I can make out the following information from the picture:
The "Epoch #" label on the x-axis indicates how many times the training set of data was run through the model.
Two lines are displayed on the y-axis: one for training accuracy (train_acc) and one for training loss (train_loss). The same graph most likely has validation accuracy (val_acc) and validation loss (val_loss) curves displayed on it, albeit it might be hard to notice them because of the axis label cutoff.
As more training epochs are added to the model, training loss and accuracy curves usually exhibit a declining tendency. This indicates that the model is improving its ability to match the training set.
Accuracy and validation loss curves are used to track how effectively the model generalizes to new data. Although the validation loss and accuracy curves should ideally also have a downward trend, if the model is beginning to overfit the training data, they may begin to plateau or even slightly increase.
Regretfully, there isn't enough information in the image you gave to say with certainty how well the DenseNet model is working with the corrosion dataset.
To reach a more conclusive decision, we would also need to observe the following other items:
The y-axis values are: This would enable us to view the accuracy values and the extent of the loss.
The accuracy and loss curves for validation: These curves may indicate that the model is overfitting if they are moving upward.

Convolutional neural networks with the DenseNet architecture have demonstrated efficacy in image classification tasks overall. More data would be required to determine whether a DenseNet model will work well for corrosion detection on this dataset.

4.4.ResNet

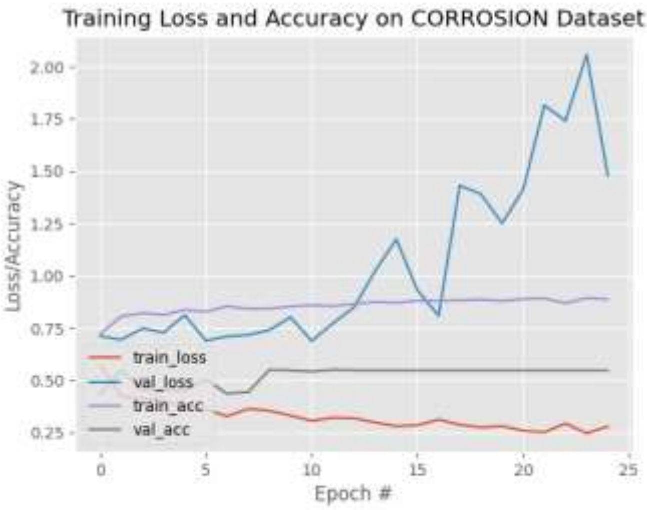

Figure 11. Line graph depicting results of ResNet.

The accompanying graphic is a line graph that illustrates a ResNet model's accuracy and training loss using a corrosion dataset. Though the text label on the y-axis is partly obscured, it most likely reads "Loss/Accuracy" according to standard corrosion detection graphs.
I can make out the following information from the picture:
The "Epoch #" label on the x-axis indicates how many times the training set of data was run through the model.
Two lines are displayed on the y-axis: one for training accuracy (train_acc) and one for training loss (train_loss). The same graph most likely has validation accuracy (val_acc) and validation loss (val_loss) curves displayed on it, albeit it might be hard to notice them because of the axis label cutoff.
As more training epochs are added to the model, training loss and accuracy curves usually exhibit a declining tendency. This indicates that the model is improving its ability to match the training set.
Accuracy and validation loss curves are used to track how effectively the model generalizes to new data. Although the validation loss and accuracy curves should ideally also have a downward trend, if the model is beginning to overfit the training data, they may begin to plateau or even slightly increase.
Considering the trends that are clear in the graph:
As more epochs are added, the training loss (train_loss) seems to be reducing. This implies that the training data is being used to teach the model.
The number of epochs looks to be boosting the training accuracy (train_acc). This implies that as the model goes through more training rounds, it is becoming more effective on the training data.
Regretfully, there is not enough information in the image you submitted to view the validation accuracy (val_acc) and validation loss (val_loss) curves. These are essential for figuring out how well the ResNet model generalizes to new information.
To reach a more conclusive decision, we would also need to observe the following other items:
The y-axis values are: This would enable us to view the accuracy values and the extent of the loss.
The accuracy and loss curves for validation: These curves may indicate that the model is overfitting if they are moving upward.
Convolutional neural networks, such as ResNet, have been demonstrated to perform well in image classification tasks overall. With this dataset, a ResNet model might work well for corrosion detection; however, further data would be required to make a firm determination.

Comparisons of models
1.Architecture Model:

- VGG16: With 16 weight layers, VGG16 has a deep architecture. Several convolutional layers precede its completely coupled layers.

- MobileNet: Based on depth wise separable convolutions, MobileNet is a lightweight architecture designed to minimize computational complexity. Mobile and embedded devices are the design's target market.
- DenseNet: DenseNet promotes feature reuse and gradient flow throughout the network by introducing dense connections between all layers inside a dense block.
- ResNet: ResNet addresses the vanishing gradient issue by using residual connections to aid in the training of extremely deep networks.

2.Training Efficiency:

- VGG16: Because of its deep architecture and many parameters, VGG16 may overfit and require a lot of processing power to train.
- MobileNet: MobileNet is an extremely efficient use of computational resources, which makes it appropriate for implementation on devices with limited capabilities, including embedded systems and mobile phones.
- DenseNet: Faster convergence and enhanced parameter efficiency are the results of Dense Net's densely connected architecture, which encourages feature reuse and increases training efficiency.
- ResNet: ResNet can train very deep networks with superior gradient flow, which leads to faster convergence and greater generalization. ResNet does this by using residual connections.

3. Performance:

- VGG16: This model has been used extensively as a baseline for several computer vision tasks and has shown good results on benchmarks for picture classification.
- MobileNet: This model achieves good accuracy at low computational complexity by striking a compromise between model size and performance.
- DenseNet: Because of its densely connected topology, DenseNet has shown state-of-the-art performance on image classification tasks with less parameters than other networks.
- ResNet: Thanks to its capacity to train extremely deep networks efficiently, ResNet has outperformed earlier state-of-the-art models on several image recognition benchmarks, achieving astonishing results.

4. Application Specificity:

- VGG16: This model has been widely used as a foundation for various computer vision applications due to its outstanding performance on photo categorization benchmarks.
- MobileNet: This model delivers good accuracy at low computational complexity by striking a balance between model size and performance.
- DenseNet: Because of its densely connected architecture, DenseNet has outperformed other networks in image classification tasks requiring fewer parameters.
- ResNet: ResNet' s capacity to train extraordinarily deep networks effectively has allowed it to outperform earlier state-of-the-art models and provide remarkable results on several image recognition benchmarks.

In conclusion, various CNN designs are appropriate for a variety of applications and deployment circumstances because to their own advantages and shortcomings. While VGG16 and ResNet are tried-and-true models that perform well in a variety of applications, MobileNet and DenseNet offer customized solutions that are geared for efficiency and feature reuse, respectively. The type of work, the computational resources available, and the performance requirements all play a role in the model selection.

V. FUTURE PROSPECTS

1. Enhanced Model Performance: Continued research and refinement of the CNN architectures used in the project, such as optimizing hyperparameters, exploring different network depths, or incorporating advanced techniques like transfer learning, could lead to further improvements in model performance. This could result in higher accuracy and robustness in corrosion detection across different datasets and environmental conditions.
2. Integration with Sensor Networks: Integrating the developed models into sensor networks deployed along underwater pipelines could enable real-time corrosion monitoring and early detection of potential issues. This would facilitate proactive maintenance strategies and help prevent costly pipeline failures and environmental incidents.
3. Automation and Autonomous Systems: The integration of corrosion detection models into autonomous systems, such as underwater robots or drones, could enable automated inspection and maintenance of underwater pipelines. This

would reduce the need for manual intervention, enhance operational efficiency, and improve safety by minimizing human exposure to hazardous environments.

4. Data Fusion and Multimodal Sensing: Combining data from multiple sources, such as visual imagery, acoustic signals, and environmental sensors, could provide richer contextual information for corrosion detection. This approach, known as data fusion, could enhance the reliability and accuracy of corrosion detection systems by leveraging complementary sources of information.

5. Edge Computing and Edge AI: Implementing the developed models on edge computing platforms deployed near the source of data acquisition could enable real-time processing and analysis of sensor data without relying on centralized computing infrastructure. This would reduce latency, bandwidth requirements, and dependency on cloud services, making corrosion detection systems more responsive and resilient.

6. Scalability and Generalization: Expanding the scope of the project to encompass a wider range of corrosion types, materials, and environmental conditions would enhance the scalability and generalization of the developed models. This would ensure their applicability across diverse real-world scenarios and industries beyond underwater pipelines, such as infrastructure inspection, manufacturing, and environmental monitoring.

7. Collaboration and Interdisciplinary Research: Collaborating with domain experts from fields such as materials science, corrosion engineering, and marine biology could provide valuable insights and domain-specific knowledge for refining the corrosion detection models. Interdisciplinary research efforts could lead to innovative solutions and novel applications in corrosion monitoring and mitigation.

VI. Conclusion

For this project, we have investigated the efficacy of four well-known convolutional neural network (CNN) architectures for the job of underwater pipeline corrosion detection: VGG16, MobileNet, DenseNet, and ResNet. We have learned a great deal about each model's advantages, disadvantages, and useful considerations through thorough investigation and testing.

According to our research, the four CNN architectures perform admirably when it comes to identifying corrosion in underwater pipelines. The resilience and simplicity of VGG16 and the efficiency and applicability of MobileNet for deployment on devices with limited resources are just two of the distinct advantages that each model offers. ResNet' s residual connections make it easier to train very deep networks with better generalization, while Dense Net's densely linked architecture allows for efficient feature reuse and gradient flow.

Furthermore, to achieve the best results in corrosion detection tasks, our study emphasizes the significance of data quality, preprocessing methods, and model optimization tactics. We have shown the potential for automated, real-time corrosion monitoring systems that can improve the safety, dependability, and efficiency of underwater pipes by utilizing developments in deep learning, sensor technology, and edge computing.

To improve the scalability and application of corrosion detection solutions, future research directions for this project include exploring multimodal sensing techniques, integrating with sensor networks and autonomous systems, further optimizing and refining CNN architectures, and working with domain experts.

In summary, our work adds to the expanding corpus of research in corrosion detection and highlights the revolutionary potential of deep learning in tackling pressing infrastructure issues. We can make great progress toward safer, more robust underwater pipeline networks for the good of society and the environment by utilizing CNNs and fostering interdisciplinary collaboration.

VII. REFERENCES

[1]. Kandiyoti, R. Under the sea. Eng. Technol. 2009, 4, 26–28.

[2]. Shama, A.M.; Bady, A.; El-Shaib, M.N.; Kotb, M.A. Review of leakage detection methods for subsea pipeline. In Proceedings of the 17th International Congress of the International Maritime Association of the Mediterranean, Lisbon, Portugal, 9–11 October 2017.

[3]. Veritas, N.D. Selection and Use of Subsea Leak Detection Systems. Recommended Practice Det Norske Veritas DNV-RP-F302, Veritasveien, Norway. 2010.

[4]. Brunone, B. Transient test-based technique for leak detection in outfall pipes. J. Water Resour. Plan. Manag. 1999, 125, 302–306.

[5]. Meniconi, S.; Capponi, C.; Frisinghelli, M.; Brunone, B. Leak Detection in a Real Transmission Main Through Transient Tests: Deeds and Misdeeds. Water Resour. Res. 2021, 3, e2020WR027838.

[6]. Murvay, S.P.; Silea, I. A survey on gas leak detection and localization techniques. J. Loss Prev. Process Ind. 2012, 6, 966–973.

[7]. Mahmutoglu, Y.; Kadir, T. Received signal strength difference-based leakage localization for the underwater natural gas pipelines. Appl. Acoust. 2019, 153, 14–19.

[8]. Adegboye, A.M.; Fung, W.K.; Karnik, A. Recent advances in pipeline monitoring and oil leakage detection technologies: Principles and approaches. Sensors 2019, 11, 2548.
[9]. Zemel, R.S. A Gradient-Based Boosting Algorithm for Regression Problems. In Proceedings of the 13th International Conference on Neural Information Processing Systems, Hong Kong, China, 3–6 October 2006
[10]. Boaz, L.; Kaijage, S.; Sinde, R. An overview of pipeline leak detection and location systems. In Proceedings of the 2nd Pan African International Conference on Science, Computing and Telecommunications (PACT 2014), Arusha, Tanzania, 14–18 July 2014; IEEE: Piscataway, NJ, USA, 2014.
[11]. Jia, Z.; Ren, L.; Li, H.; Sun, W. Pipeline Leak Localization Based on FBG Hoop Strain Sensors Combined with BP Neural Network. Appl. Sci. 2018, 8, 146.
[12]. Manekiya, M.H.; Arulmozhivarman, P. Leakage detection and estimation using IR thermography. In Proceedings of the 2016 International Conference on Communication and Signal Processing (ICCSP), Melmaruvathur, India, 6–8 April 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 1516–1519.
[13]. Recommended, Practice. DNVL-RP-F302—Edition April 2016. Available online: https://rules.dnvgl.com/ docs/pdf/DNVGL/RP/2016-04/DNVGL-RPF302.pdf (accessed on 15 February 2019).
[14]. Datta, S.; Sarkar, S. A review on different pipeline fault detection methods. J. Loss Prev. Process Ind. 2016, 41, 97–106.
[15]. Ai, C.; Zhao, H.; Ma, R.; Dong, X. Pipeline damage and leak detection based on sound spectrum LPCC and HMM. In Proceedings of the Sixth International Conference on Intelligent Systems Design and Applications (ISDA'06), Jinan, China, 16–18 October 2006; IEEE: Piscataway, NJ, USA, 2006; pp. 829–833.
[16] Arzaghi E, Abbassi R, Garaniya V, Binns J, Chin C, Khakzad N, Reniers G (2018) Developing a dynamic model for pitting and corrosion-fatigue damage of subsea pipelines. Ocean Eng 150:391–396. https://doi.org/10.1016/j.oceaneng.2017.12.014
[17] Baldi P, Brunak S, Chauvin Y, Andersen CA, Nielsen H (2000) Assessing the accuracy of prediction algorithms for classification: an overview. Bioinformatics 16(5):412–424
[18] E. Fleming, *Construction Technology: An Illustrated Introduction*, Blackwell Publishing Ltd, Hoboken, NJ, USA, 2005.
[19]F. Bonnin-Pascual and A. Ortiz, "Corrosion detection for automated visual inspection, developments in corrosion protection," in *Developments in Corrosion Protection*, pp. 620–632, IntechOpen, London, UK, 2014.View at: Publisher Site | Google Scholar
[20]V. Bondada, D. K. Pratihar, and C. S. Kumar, "Detection and quantitative assessment of corrosion on pipelines through image analysis," *Procedia Computer Science*, vol. 133, pp. 804–811, 2018.View at: Publisher Site | Google Scholar
[21]L. Liu, E. Tan, Y. Zhen, X. J. Yin, and Z. Q. Cai, "AI-facilitated coating corrosion assessment system for productivity enhancement," in *Proceedings of the 2018 13th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, pp. 606–610, Munich, Germany, May 2018.View at: Publisher Site | Google Scholar
[22]D. J. Atha and M. R. Jahanshahi, "Evaluation of deep learning approaches based on convolutional neural networks for corrosion detection," *Structural Health Monitoring*, vol. 17, no. 5, pp. 1110–1128, 2018.View at: Publisher Site | Google Scholar
[23]S. Dorafshan, R. J. Thomas, and M. Maguire, "Comparison of deep convolutional neural networks and edge detectors for image-based crack detection in concrete," *Construction and Building Materials*, vol. 186, pp. 1031–1045, 2018.View at: Publisher Site | Google Scholar
[24]Y. Jung, H. Oh, and M. M. Jeong, "An approach to automated detection of structural failure using chronological image analysis in temporary structures," *International Journal of Construction Management*, vol. 19, no. 2, pp. 178–185, 2019.View at: Publisher Site | Google Scholar
[25]J.-M. Maatta, J. Vanne, T. Hamalainen, and J. Nikkanen, "Generic software framework for a line-buffer-based image processing pipeline," *IEEE Transactions on Consumer Electronics*, vol. 57, no. 3, pp. 1442–1449, 2011.View at: Publisher Site | Google Scholar
[26]D. Itzhak, I. Dinstein, and T. Zilberberg, "Pitting corrosion evaluation by computer image processing," *Corrosion Science*, vol. 21, no. 1, pp. 17–22, 1981.View at: Publisher Site | Google Scholar