DenseNet-Based Deep Learning Driven Multi-Class Classification of Side-Scan Sonar Images for Marine Exploration

Maddukuri Srinadh[1,*], J. B. Seventline[2]

[1]Research Scholar, Department of Electrical & Electronics and Communication Engineering, GITAM (Deemed to be University), Visakhapatnam, Andhra Pradesh, India

[2]Professor, Department of Electrical & Electronics and Communication Engineering, GITAM (Deemed to be University) Visakhapatnam, Andhra Pradesh, India

Emails: smadduku@gitam.in; sjoseph@gitam.edu

Abstract

This paper discusses about implementing Machine Learning Models with the Marine_Pulse dataset. This is about side-scan sonar images that have four groups: Engineering Platform (EP), Pipeline/Cable (P/C), Sea Bed Surface (SBS), and Underwater Residual Mound (URM). This manuscript performed some difficult feature extraction and classification methods using the DenseNet-DNN framework. This paper delves deeply into the implementation of the DenseNet121 Dropout, DenseNet201 Dropout, DenseNet201 Enhanced Dropout, and DenseNet201 Transfer Learning models. It investigates how these models perform on feature extraction and classification using a DNN. We enhanced the performance and reduced overfitting by applying a dropout to DenseNet121 and DenseNet201 and by adding transfer learning (TL) to DenseNet201, respectively. The models were evaluated based on the accuracy, precision, recall, F1-score, specificity, and classification errors of the training and testing samples. We observed that DenseNet201 Enhanced Dropout outperformed the other models, achieving the highest accuracy of 95.79%. DenseNet201 Dropout followed this achievement with an accuracy of 94.74% and DenseNet121 Dropout with an accuracy of 92.11%. DenseNet201 Transfer Learning, on the other hand, had the worst accuracy (92.11%). Specificity is a measure of how well the model represents negative examples correctly. The maximum specificity was observed in DenseNet201 Enhanced Dropout (98.38%). DenseNet201 Dropout follows it at 97.96% and DenseNet121 Dropout at 97.18%. The smallest specificity was reported on DenseNet201 TL with 96.60%. This result demonstrates that our keys can generalize well and that they maintain high classification accuracy on the test data.

Keywords: Side Scan SONAR Images; DenseNet121; DenseNet201; Transfer Learning; Deep Neural Network (DNN); Multi Class Classification

1. Introduction

The development of new sensing technologies such as side-scan sonar has revolutionized the way we explore and monitor underwater environments by allowing us to have a high-resolution image of the seafloor and other objects beneath the water. One relevant dataset is the Marine Pulse, which is suitable for analyzing side-scan sonar recordings. The features inside this corpus offer useful hints about the undersea background, such as emergent classes, showing different seafloor features, marine elements, or sound phenomena. Side-scan sonar (SSS) is a category of sonar system that is used to create an image of the ocean floor, even when the water is murky, objects are camouflaged, and structures at the sea floor are indistinct due to lack of light. It works by transmitting sound waves to the sides of a sonar system, which is either dragged through the water or installed in a stationary position. Side-scan sonar detects

DOI: https://doi.org/10.54216/JCIM.160213
Received: January 06, 2025 Revised: March 05, 2025 Accepted: April 02, 2025

echoes of these sound waves to generate detailed images. These images have several different uses, including mapping underwater terrain, examining archaeology, monitoring habitats, and aiding in search-and-recovery operations. Side-scan sonar systems are generally employed in several setups, such as hull-mounted sonars (HMS) towed by a towfish or pole-mounted sonars (SSS) positioned at the front or the side of the boat. They are made up of two linear projectors that generate short sound pulses in a broad beam, which intercept the sea bottom in narrow strips [4]. The SSS system consists of parts called a transducer, which transmits high-frequency sound waves; a receiver, which receives the reflected sound waves; and a control unit, which processes the signals to be visualized.

Image quality is impacted by sonar frequency; manufacturers raise frequencies to achieve higher resolution and finer roughness. Increased false reflection features, however, may make it more difficult to distinguish objects underwater. Horizontal beamwidth, towing speed, operational range, and ping rate all affect SSS's along-track resolution, whereas pulse length, water sound speed, and local grazing angle all affect its across-track resolution [5,6].

Manual target interpretation in SSS pictures is currently laborious, subjective, and inefficient. Using traditional machine-learning techniques [7], researchers have investigated automatic detection techniques; nevertheless, they encounter difficulties when dealing with complex terrain imagery. Although deep learning approaches, especially deep CNNs [8], have demonstrated promise for underwater identification, they are beset by issues such as intricate structures, poor performance, and underrepresentation of target samples because of expensive data acquisition and sluggish collection rates [9].

The key features About Side-Scan Sonar images.

i. ***Clear Pictures*:** Can get small details of the ocean floor stuff.

ii. ***Big Area Scan*:** Can look at big spots on the bottom in one go, good for big surveys.

iii. ***Shadow Look***: Things make shadows in the pictures, giving hints about how they look and size.

The Marine Pulse images dataset [10] was created with the help of side-scan sonar technology. They create grayscale images that depict the form and composition of the seafloor by recording sound waves that reflect off underwater surfaces. Accurately comprehending and classifying these visuals is crucial. Numerous fields, including underwater archaeology, environmental monitoring, marine life research, and military operations, can benefit from this.

The research aims to classify side-scan sonar pictures from the Marine Pulse dataset, addressing challenges like sharpness changes, background noise, and overlap parts. In this paper, we are using DenseNet architectures for feature extraction. The main advantages using this are

i. DenseNet structures provide direct connections between two layers of the same feature-map size, which lessens the problem of vanishing gradients and aids in feature reuse. Better learning and model performance are the outcomes of this.

ii. Because DenseNets do not have to train duplicate feature maps, they require fewer parameters than standard convolutional networks. When dealing with complicated data sets, such as side-scan sonar images, this is extremely helpful.

iii. Study say DenseNet models, as if DenseNet121 and DenseNet201 are better than InceptionV3, Xception, and MobileNetV2 for classifying sonar pictures right.

The article is structured in sections as: Section 2 examines related research contributions and provides a brief summary of recent studies. Section 3 provides the architecture of proposed framework employed. Section 4 discusses the computational analysis and gives the findings; Section 5 summarizes the findings and proposes areas for future investigation.

2. Related Work

This paper [11] demonstrates the extreme scenario in which a DNN can identify objects in single-shot super-resolution (SSS) photos without prior training data. This constitutes the "0-shot learning problem." The authors employed a pseudo-modeling strategy for a cluster to modify the model with existing optical and SSS pictures, subsequently training the DNN with these models. Consequently, the 0-shot learning challenge can be reconfigured as a conventional learning maintenance issue. Experimental findings indicate that optimal resource allocation is attained even in the absence of training examples.

DOI: https://doi.org/10.54216/JCIM.160213
Received: January 06, 2025 Revised: March 05, 2025 Accepted: April 02, 2025

This research [12] proposes a unique deep learning model utilizing an adaptive weighted CNN for the classification of underwater sonar pictures. To address the issue of initial inadequacy of weighted filters in convolutional neural networks (CNN), the implementation of AW-CNN utilizes weights produced by deep belief networks (DBN) to substitute the filters acquired in CNN. Underwater sonar image classification makes use of the weights. First, the authors use the dimension function to perform the dimension transformation and concatenate the inputs of CNN and DBN. Based on the dimensional change, the authors synthesize two models and modify the random training filter weights of the CNN. A local response normalization (LRN) operation is formulated to normalize the weight changes in the original network to promote classification accuracy. The results show that the new AW-CNN method can easily and effectively use sonar images to identify seabed types better than current methods, which helps in exploring oil, gas, and minerals in shallow and deep waters.The work[13] proposes a new target detection method for SSS picture recognition of targets that employs Neutral Set (NS) and Diffusion Map (DM). In the first, an input SSS image is transformed into the NS space to acquire the NS image subset. Secondly, we apply a gray threshold method to the SSS picture to define the shadow area, prior to computing the diffusion map. We then learn the target equation characterized by intensity and texture difference through the diffusion map to define the target region. One can see that the proposed algorithms can serve as a feasible scheme for SSS images with clear or nonclear targets and shadows. Experiments utilizing SSS photos of diverse targets (both with and without shadows) shown that no genuine or imperceptible effects transpire. The target area can be accurately identified in SSS pictures exhibiting intricate features like sand dunes. The proposed algorithm's accuracy and efficacy are assessed.

This work [14] introduces a methodology that integrates adaptive learning with deep learning SSS image classification. Deep learning enhances classification accuracy and employs adaptive learning to address issues where deep neural networks are constrained by insufficient standard training. The researchers developed a pre-trained CNN for image classification tasks using significant training data, and then optimized it especially for side-scan sonar image classification tasks, affecting transfer efficiency. Experiments show that this method may reduce overfitting in deep neural networks with few data while retaining high classification accuracy.

This article[15] reviews some of the existing researches from 2015 to 2021 that use various techniques such as deep CNN, machine learning, and image processing to identify underwater noise in sonar images. It also examines how to identify underwater objects as mines, rocks, mud, and other non- mine objects. Also, many researchers have developed numerous CNN architectures for underwater detection and segmentation. The summary provides a big look at what each author's research area and contribution is, like when they published it, what methods they used, goals, how they gathered info, shared it, and types of events.

Automatic target recognition improves target planning and distribution to autonomous management systems for application security and protection. The support vector machine (SVM) is a well-established classification model technique. They provide the maximum marginal distribution without overfitting the data. The selection of kernel function is well recognized to enable the utilization of particular data inside the distribution. This paper [16] shows that for classification in SSSs, supplemental feature extraction and data architecture provide better results than optimization alone.

SSS is extensively employed in Submarine operations for rescue and the identification of target elements at sea, including vessels and airplanes. Nonetheless, the implementation of automatic target categorization in side-scan sonar imagery remains inadequate due to the scarcity of data sets featuring certain structural traits and the limited dimensions of the images. Secondly, the actual data in side-scan sonar images is imbalanced. This work [17] provides a side-scan sonar image segmentation approach utilizing synthetic data and transfer learning. This method employs optical pictures as input and produces "simulated side-scan sonar images" through network transformation models to replicate side-scan sonar imagery. This study demonstrates via experiments that optimizing the learning of a pre-trained neural network using a training set of "generated SSSs"achieves an accuracy of up to 97.32%. Results show that a pre-trained convolutional neural network coupled with side-scan sonar-like images is effective for improving the classification performance.

The classification of anthropogenic seabed items between 3–23 kHz in SSS images has particular problems, primarily because of the complexity and variability of the underwater environment. Traditionally, experts turn to classical machine learning algorithms in conjunction with handcrafted tools for the manual interpretation of the SSS image. Although convolutional neural networks (CNNs) improve classification in this area, they struggle with different seabed textures like rocks or uneven sand. Recently, Visual Transformers (ViTs) have shown promise to mitigate these restrictions using self-supervised learning to capture global information in patches of images and therefore facilitate more flexibility to process the order of rows. In this work [18], the effectiveness of the ViT model is

DOI: https://doi.org/10.54216/JCIM.160213
Received: January 06, 2025 Revised: March 05, 2025 Accepted: April 02, 2025

rigorously compared with widely used CNN architectures such as ResNet and ConvNext on SSS image binary classification tasks. These data include a wide range of regional seabed types and balance the presence and absence of artefacts. The ViT-based model shows good classification metrics in terms of f1-score, accuracy, recall and precision, although it costs more operating money. CNNs have inductive biases and exhibit high computing efficiency and are appropriate for usage in constrained situations such as underwater vehicles. Future research can be done by exploring the combination of self-monitoring learning and ViT to improve performance in challenging underwater environments.

As deep learning advances, CNNs are progressively employed for image recognition and underwater object classification in marine research. Employing CNN to discern targets from SSS imagery can enhance recognition precision and accuracy. The plethora of available CNN models complicates the selection of an appropriate model for object recognition in SSS imagery. This paper [19] examines the predicted accuracy and performance of numerous CNN algorithms. Four traditional CNN models were initially utilized for training and predicting the same undersea SSS dataset, employing both the original and modified learning models. The authors analyze and assess the predictive accuracy as well as efficiency of four convolutional neural network models. The results demonstrate that the process of learning improves the accuracy of all CNN models, with slight increases observed for AlexNet and VGG-16, and more significant improvements noted for GoogleNet and ResNet101. GoogleNet demonstrates exceptional predictive performance, attaining 100% accuracy on training data and 94.27% on test data, while also improving computational efficiency.

Self-tracking has demonstrated efficacy in facilitating the comprehension of visual representations without reliance on extensive texts. The advancement of computer vision algorithms to enhance comprehension, such as sonar image categorization for underwater robotics, is crucial. The privacy of sonar images and the challenges in understanding them hinder the development of extensive, publicly available sonar datasets for training supervised learning algorithms. In this study [20], the authors examined the efficacy of our self-supervised methodologies (RotNet, denoising autoencoder, and Jigsaw) in acquiring optimal picture categorization representations autonomously. They present the results of pre-training and transform learning on a dataset of real sonar images. The findings showed that the distribution of pre-training self-monitoring for all three approaches was comparable to pre-training monitoring across many learning variables.

The image processing procedure for object identification includes intermediary phases such as noise filtering, edge detection, and segmentation. Edge detection is a fundamental phase in image segmentation and object detection. This paper [21] contrasts fuzzy logic techniques with blocking methods for edge detection. The summary analysis of the results indicates that the blind spot method outperforms the fractional block method in detecting objects and shadows in sonar images.

This study [22] examined several categories of drowned individuals, ships, airplanes, mines, and seabed as depicted in sonar imagery. Initially, a comprehensive real-time side-scan sonar picture collection, SeabedObjects-KLSG, was developed through extensive accumulation, comprising 385 shipwreck images, 36 images of deceased individuals, 62 aircraft photographs, 129 ship images, mines, and 578 beach photographs. Secondly, they suggested a semi-synthetic data generation approach to produce sonar images of aircraft and submerged individuals, utilizing feedback-based optical images and segmented images from multiple regions, while addressing the discrepancies present in actual data. Ultimately, by employing pre-trained deep convolutional neural networks such as VGG19 as well as fine-tuned model with 70% of the authentic data alongside semi-synthetic data, the remaining 30% of the authentic data may be enhanced to 97.76%, representing the highest performance across all methodologies. This work illustrates that integrating deep learning with semi-synthetic profile creation is a viable method to enhance the accuracy of underwater object detection.

A major concern in today's marine applications is to create more space by eliminating the large numbers associated with the estimates. It may be impossible to fill this space with a reasonable number of illustrative examples. Consequently, it is essential to diminish the residual feature space by either modifying the data (e.g., employing principal component analysis) or by judiciously eliminating features. The authors [23] recommend that dimensionality reduction be a priority in all aspects of the study. They also recommend using the information collection recommendations prepared for field surveying. Among these, one can find an error matrix, confidence bounds, and two or more measures of accuracy (global and for a particular group).

In this paper [24], the authors focus on the in-depth study of deep learning based on both above and beneath the water objects knowledge. To facilitate a comprehensive review, the authors first summarize the basic concepts and design principles in a unified manner. Therefore, they collected all popular/standardized datasets used for marine object

DOI: https://doi.org/10.54216/JCIM.160213
Received: January 06, 2025 Revised: March 05, 2025 Accepted: April 02, 2025

recognition and performed deep learning by comparing the depths. Furthermore, the results and future models on marine object recognition are discussed in depth. Consequently, they attain advanced marine object understanding through the application of deep learning techniques.

This paper [25] presents a self-localization algorithm for a side-scan sonar-based underwater vehicle (AUV). Machine vision continuously provides a visual representation of multiple clusters derived from dual-frequency sonar images, subsequently facilitating the generation of small-scale bathymetric maps from these images. The differences in feature clusters, particularly in feature sizes and features acquired at various sonar operating frequencies, enable efficient multimodal matching of characteristics. The comparison of newly acquired features with the previously utilized map can yield highly efficient solutions in real time. Feature matching is conducted in four dimensions for each feature class to establish a correlation between feature location and perceptual strength. All performance and regulatory requirements have been validated using real-world sonar imagery obtained from the Iver3 AUV platform.

SSS has proven effective for seabed scanning tasks; however, sonar images produced by SSS often suffer from noise and geometric distortions, which can affect the perceived texture, smoothness, size, and shape of seabed objects. Therefore, for the detection of underwater objects, it is important to first reduce the noise and reconstruct the real image. This study presents a method for the correction and reconstruction of sonar images, which includes intensity normalization, multiple tilt correction, optical and acoustic correction, as well as velocity and location correction. This is accomplished through the utilization of navigation and inertial data acquired from the sensors of autonomous underwater vehicles.

This paper [27] uses GANs to generate realistic sonar images. After the ray-tracing-based sonar simulator calculates semantic information from the scene, the GAN-based style modification algorithm generates genuine sonar images. The authors assessed approach by comparing real-world sonar images to ground-truth photos of various items. The deep object detection system is appropriate for underwater activities including automatic shipwreck search, minesweeping, and landmark-based navigation. Experimental findings show that the suggested method can generate accurate target object training images. The technique generates accurate maps from numerous perspectives and situations without sea trials, making it easy and powerful. Other sonar-based algorithms and object detection can employ the suggested realistic sonar picture generation method.

3. Research Gaps

The study employs the Marine Pulse dataset, which consists of side-scan sonar images, to investigate multi-class classification. In order to improve classification accuracy and computational efficiency, DenseNet121 and DenseNet201 are utilized for feature extraction and combined with a Deep Neural Network (DNN) classifier. SONAR imaging tasks can benefit from DenseNet models since they promote feature reuse and lessen vanishing gradients.

The existing CNN models like Xception, InceptionV3, and MobileNetV2 often do not perform well in sonar image tasks, while DenseNet models provide better results by encouraging feature reuse and reducing vanishing gradients.

A work [28] compared CNN architectures and showed how models like Xception and MobileNetV2 have limits in specialized areas, especially with side-scan sonar image datasets. Furthermore, the paper [29] presented effective DenseNet variations for remote sensing, but their use in sonar imagery has not been explored. Likewise, the paper [30] focused on binary classification in side-scan sonar target detection, leaving a notable gap in multi-class classification issues. Our objective is to implement DenseNet121 and DenseNet201 with DNN Classifier to successfully classify the dataset consisting of sonar imagery

Computational efficiency is another key factor. DenseNet architectures use fewer parameters, lowering memory use and computational demands, making them suitable for resource-limited settings. In contrast, models like InceptionV3 need more substantial hardware resources. In conclusion, our study aims for unambiguous models and reasoning while leveraging DenseNet's advantages in feature reuse, efficiency, and fit for sonar imaging challenges. Our approach overcomes problems in previous studies and delivers the best results in side-scan sonar image classification by concentrating on multi-class classification and including effective design and clarity.

The following research gaps identified and addressed in this paper

1. Limited Evaluation of Sonar imagery Data: A lot of research examines broad picture problems without paying enough attention to particular datasets, such as side-scan sonar images.

2. Model Understanding: Deep learning models are opaque, which makes it challenging to comprehend and believe their predictions, especially in crucial domains like undersea protection and exploration.

DOI: https://doi.org/10.54216/JCIM.160213
Received: January 06, 2025 Revised: March 05, 2025 Accepted: April 02, 2025

3. Memory and Resource Limitations: Certain models require a lot of processing power, which makes them challenging to apply in settings with constrained resources.

4. Multi-Class Classification Challenges*:* Some studies have solely looked at binary classification, ignoring the challenges associated with multi-class classification in sonar pictures.

4. Proposed Framework

The Marine_Pulse dataset has side-scan sonar images arranged into four classes—Engineering Platform (EP), Pipeline/Cable (P/C), Sea Bed Surface (SBS), and Underwater Residual Mound (URM)—which is the basis for the classification task.

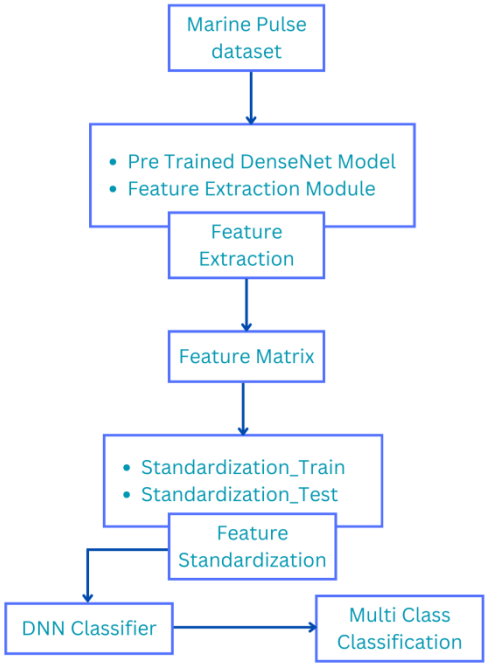

Figure 1. Proposed DenseNet-DNN Framework

The process begins with input sonar images, represented mathematically as an input vector

$$\mathbf{X} \in \mathbb{R}^{m \times n \times c}$$

Where *m* and *n* are the image dimensions, and *c* is the number of channels.

Feature Extraction with Pre-trained DenseNet Models

To extract meaningful features from the images, pre-trained DenseNet models (e.g., DenseNet121 and DenseNet201) are employed. These models, represented as a function

$$f_{\text{DenseNet}}: \mathbb{R}^{m \times n \times c} \rightarrow \mathbb{R}^{d} \quad (1)$$

map the high-dimensional input images to a lower-dimensional feature space. The output of this stage is a feature vector

$$\mathbf{F} = f_{\text{DenseNet}}(\mathbf{X}) \quad (2)$$

DOI: https://doi.org/10.54216/JCIM.160213
Received: January 06, 2025 Revised: March 05, 2025 Accepted: April 02, 2025

Where$\mathbf{F} \in \mathbb{R}^d$and d is the dimensionality of the feature space. This feature vector encapsulates the most relevant patterns and characteristics required for classification while discarding redundant information.

Feature Matrix and Standardization

The feature vectors extracted from all input images are assembled into a feature matrix$\mathbf{M} \in \mathbb{R}^{N \times d}$, where N is the total number of images. To ensure consistent feature scaling and improve model performance, standardization is applied separately to the training and testing feature matrices. For training data, the standardized feature matrix is given by:

$$\mathbf{M}_{\text{train}}^{\text{std}} = \frac{\mathbf{M}_{\text{train}} - \mu_{\text{train}}}{\sigma_{\text{train}}} \quad (3)$$

Where μ_{train} and σ_{train} are the mean and standard deviation of the training feature matrix. Similarly, for testing data:

$$\mathbf{M}_{\text{test}}^{\text{std}} = \frac{\mathbf{M}_{\text{test}} - \mu_{\text{train}}}{\sigma_{\text{train}}} \quad (4)$$

This ensures that the testing data is standardized using the same parameters as the training data.

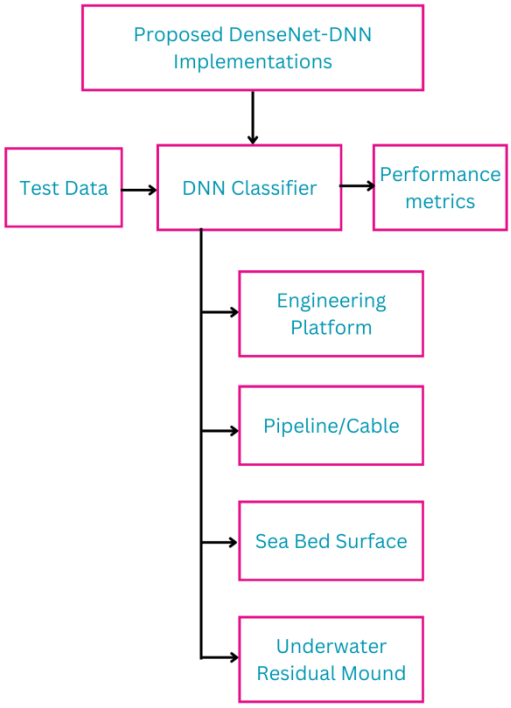

Figure 2. DNN Classifier for Multi Class Classification

The standardized feature matrix $\mathbf{M}_{\text{test}}^{\text{std}}$ serves as input to the Deep Neural Network (DNN) classifier, which maps the feature vectors to class probabilities. The DNN can be represented as a function

$$f_{\text{DNN}}: \mathbb{R}^d \rightarrow \mathbb{R}^C \quad (5)$$

where C is the number of classes. The classifier outputs a probability distribution for each input feature vector $\mathbf{F}_i$, defined as:

$$\mathbf{P}_i = f_{\text{DNN}}\left(\mathbf{F}_i^{\text{std}}\right) \quad (6)$$

Where$\mathbf{P}_i \in \mathbb{R}^C$and $\sum_{j=1}^{C} P_{ij} = 1$.

The predicted class for an input vector is obtained as:

DOI: https://doi.org/10.54216/JCIM.160213
Received: January 06, 2025 Revised: March 05, 2025 Accepted: April 02, 2025

$$\hat{y}_i = \text{argmax}_j \mathbf{P}_{ij} \quad (7)$$

where $\hat{y}_i$ is the predicted class label for the i^{th} image.

Transfer learning leverages pre-trained models to adapt knowledge from a source domain$\mathcal{D}_S$to a target domain $\mathcal{D}_T$, where $\mathcal{D}_S \neq \mathcal{D}_T$. The goal is to transfer feature representations learned on a large dataset (e.g., ImageNet) to improve performance on a different but related task.

1. *Notation and Definitions*

Source domain is represented as in eqn. 8

$$\mathcal{D}_S = \{(x_i^S, y_i^S)\}_{i=1}^{N_S} \quad (8)$$

Where

$x_i^S \in X_S$ are source domain images.

$y_i^S \in Y_S$ are corresponding labels.

$P(X_S, Y_S)$ is the joint probability distribution.

Target domain is represented as in eqn. 9

$$\mathcal{D}_T = \{(x_i^T, y_i^T)\}_{i=1}^{N_T} \quad (9)$$

where

$x_i^T \in X_T$ are target domain images.

$y_i^T \in Y_T$ are corresponding labels.

$P(X_T, Y_T)$ is the joint probability distribution as written in eqn. 10. In transfer learning,

$$P(X_S, Y_S) \neq P(X_T, Y_T) \quad (10)$$

But the feature space is similar.

2. Feature Extraction in Transfer Learning

A deep neural network, DenseNet201is pre-trained on the source domain $\mathcal{D}_S$, learning a feature representation function represented in eqn. 11

$$Z_S = f_{\theta_S}(x^S) \quad (11)$$

Where,

f_{θ_S} is a convolutional feature extractor (pre-trained layers).

Z_S represents extracted features.

For the target domain, the same pre-trained feature extractor is used as in eqn. 12

$$Z_T = f_{\theta_S}(x^T) \quad (12)$$

where θ_S is frozen (not trainable) or fine-tuned (partially trainable).

3. Transfer Learning via Fine-tuning

Fine-tuning adapts the model by unfreezing some layers and updating weights represented in eqn. 13

$$\theta_T = \theta_S + \Delta\theta \quad (13)$$

Where,

$\theta_T$are the new fine-tuned parameters.

$\Delta\theta$represents small updates from target domain training.

Only the last few layers (e.g., fully connected layers) are newly trained represented in eqns. 14, 15, 16

DOI: https://doi.org/10.54216/JCIM.160213
Received: January 06, 2025 Revised: March 05, 2025 Accepted: April 02, 2025

$$h_1 = \text{ReLU}(W_1 Z_T + b_1) \quad (14)$$

$$h_2 = \text{ReLU}(W_2 h_1 + b_2) \quad (15)$$

$$\hat{y} = \text{softmax}(W_3 h_2 + b_3) \quad (16)$$

Where,

W_1, W_2, W_3 are trainable parameters for the target domain.

4. *Transfer Loss Function*

The loss function used in transfer learning is typically Categorical Cross-Entropy Loss is given by eqn. 17

$$\mathcal{L}_{\text{transfer}} = -\sum_{i=1}^{N_T} \sum_{c=1}^{C} y_{i,c}^{T} \log\left(\hat{y}_{i,c}^{T}\right) \quad (17)$$

Where,

$y_{i,c}^{T}$ is the true one-hot encoded label.

$\hat{y}_{i,c}^{T}$is the predicted probability for class cc.

The total loss includes both feature learning and classification is given by eqn. 18

$$\mathcal{L} = \lambda \mathcal{L}_{\text{feature}} + (1 - \lambda)\mathcal{L}_{\text{classifier}} \quad (18)$$

Where,

$\mathcal{L}_{\text{feature}}$is the knowledge transferred from pre-trained layers.

$\mathcal{L}_{\text{classifier}}$ is the newly trained classifier loss.

λ is a weighting factor for balancing feature reuse and new learning.

5. Optimization

The model is optimized using Adam is as represented in eqn.

$$\theta^{(t+1)} = \theta^{(t)} - \alpha \cdot \frac{\partial \mathcal{L}_{\text{transfer}}}{\partial \theta} \quad (19)$$

Where,

α is the learning rate.

$\theta$are the trainable parameters.

This approach allows knowledge from a large dataset to improve performance on a smaller dataset.

5. Computational Analysis

A. *Dataset*

B. The Marine_Pulse dataset is a collection of side-scan sonar images made for figuring out marine exploration. It has four different classes showing various underwater structures and surfaces, giving a big setup for looking at marine places. The dataset is set up to help with classifying images, especially with side scan sonar image technology.

DOI: https://doi.org/10.54216/JCIM.160213
Received: January 06, 2025 Revised: March 05, 2025 Accepted: April 02, 2025

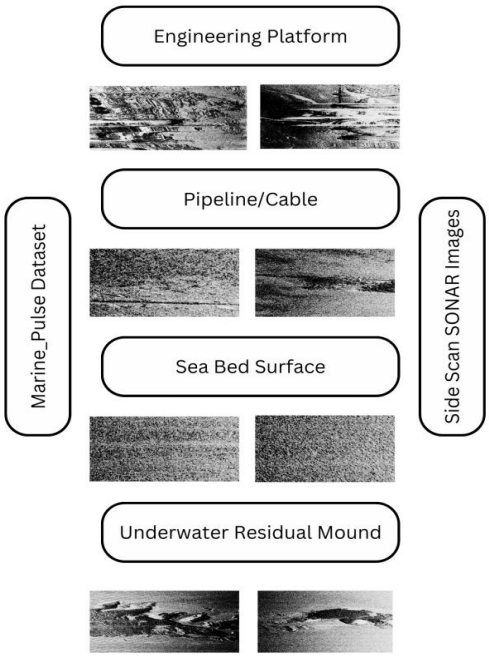

Figure 3. Marine_Pulse Dataset

The classes in Marine_Pulse dataset are

i. ***Engineering Platform (EP)***: Shows images of built things by human, like oilrigs or structures in the ocean that are usually seen underwater. These images show special shapes and building parts that show they were made with design plans

ii. ***Pipeline/Cable (PC)***: Has images of things like pipes or wires in water, usually seen with long and straight shapes. These features are very important for keeping track of what is under the water and making sure it is working right.

iii. ***Sea Bed Surface (SBS)***: Holds the images of ocean floor content with different looks and sandy designs. These images are useful for figuring out how the sea bottom is shaped and how mud gathers on the ground underwater

iv. ***Underwater Residual Mound (URM):*** Shows images of leftover bumps or heaps that come from digging underwater or natural dirt building up. These images have strange shapes and bumpy surfaces.

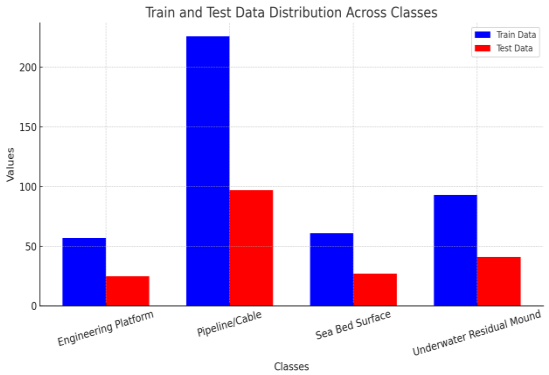

Figure 4. Marine_Pulse Dataset Distribution

DOI: https://doi.org/10.54216/JCIM.160213
Received: January 06, 2025 Revised: March 05, 2025 Accepted: April 02, 2025

The graph # 1 shows distribution of images in each class for training and testing:

i. Pipeline/Cable group is with most images in both sets(train and test), reflecting its importance

ii. The Engineering Platform and Sea Bed Surface groups have less numerous images but show balance between training and testing.

iii. The Underwater Residual Mound group has some images, giving a fair amount for both training and testing.

The Marine_Pulse dataset is good for implementation of machine learning models that classify marine features. There are many different images in the four classes, which makes it a solid test area for knowing underwater shapes. This can help in marine exploration, keeping an eye on buildings, and studying the environment, including both human-made things and natural elements makes sure the dataset is fair and true-to-life for sonar image classification

The observation and analysis of the training and testing values is important to check how well a machine learning model works. These values indicate how good the model is at learning from the training data and if it can deal with new data. By looking at these figures, one can see if the model is not learning enough, learning too much, or doing well.

Training Values Analysis looks at how the model acts on the training data. Numbers like accuracy (train and val), loss (train and val) over different epochs tell us how good the model is at finding patterns in the data it has seen. A high training accuracy and low loss imply that the model has learned a lot from the data.

Testing Values Analysis evaluates the performance of the model on the test data. Quantities such as testing accuracy, precision, recall, F1-score, and confusion matrices illustrate when and how the model excels or flounders across classes. These numbers are important to learn more about whether the model is able to generalize to real-world data when dealing with unseen samples.

In conclusion, training and testing metrics provide a comprehensive overview of the model's performance and guide informed decisions on how to further improve it. This section goes deeper into examining these metrics to identify patterns, ensure reliability, and validate the model for application in real-world situations.

C. Training Metrics Analysis

This work presents a systematic method for analyzing the performance trends of parameter tuning and selecting the most promising model for subsequent applications. Each graph is displaying one of the four metrics (train_loss, val_loss, train_accuracy, val_accuracy) as a function of the number of epochs in the following four models:

Model-1-DenseNet201_Transfer Learning(DN_201_TL)

Model-2-DenseNet201 Enhanced Dropout(DN201_EDO)

Model -3 -DenseNet121 Dropout (DN_121_DO)

Model-4-DenseNet201 Dropout(DN_201_DO)

These figures provide a full picture of what, how the models learn, and how well they generalize. The train loss graph shows loss decreasing over epochs for all models using the training data. A clear drop in all models suggests good learning. Still, the speed of decrease and when it levels off vary by model. If a model levels off at a lower train loss, it indicates a good fit to the training data. However, caution is needed because very low train loss might signal overfitting if it is much lower than validation loss.

DOI: https://doi.org/10.54216/JCIM.160213
Received: January 06, 2025 Revised: March 05, 2025 Accepted: April 02, 2025

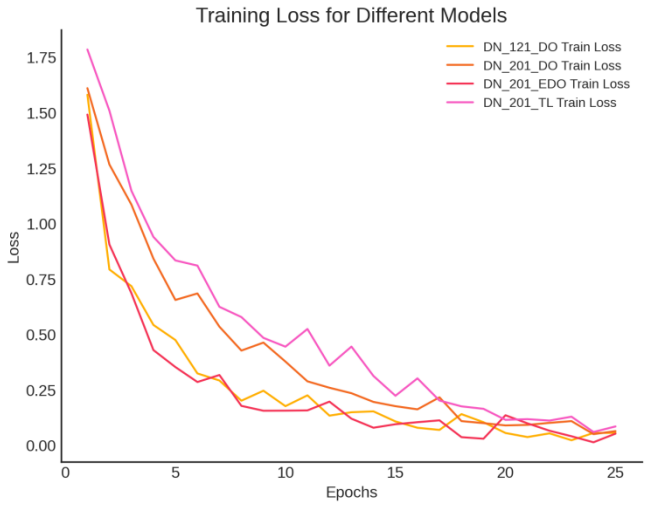

Figure 5. Train Loss Over Epochs - Comparison of all Models

The graph of validation loss shows how good each model is at unseen data. Best case, validation loss goes down with train loss and levels out eventually. Big changes in this number can show over-fitting or problems in training. Models that have validation loss curves like their train loss without big gaps show better generalization.

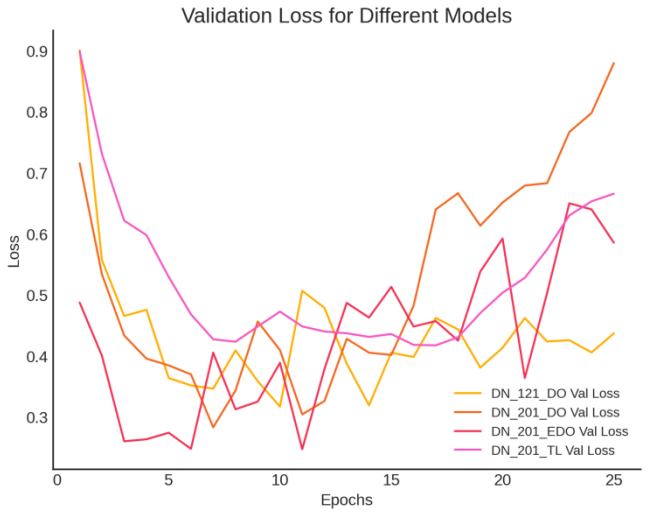

Figure 6. Val Loss Over Epochs - Comparison of all Models

The graph of train accuracy shows how well models can guess right on the training data while they learn. If accuracy goes up steadily with more epochs, that is a good thing. High final accuracy means a decent fit to the training data. However, if train accuracy is close to 100% and validation accuracy is low, then there might be a problem called overfitting.

DOI: https://doi.org/10.54216/JCIM.160213
Received: January 06, 2025 Revised: March 05, 2025 Accepted: April 02, 2025

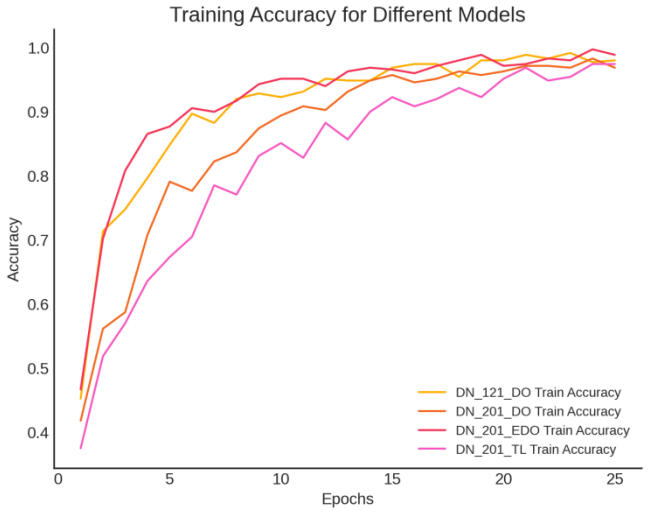

Figure 7. Train Accuracy Over Epochs - Comparison of all Models

The graph of validation accuracy shows how good models are with new data they have not seen. It is good if validation accuracy goes up over time in epochs, and if it lines up with training accuracy, that is a sign of good generalization. However, if there is a big difference between them, it might mean over-fitting is happening. If validation accuracy keeps getting better, it means the model is learning well without over-fitting too much.

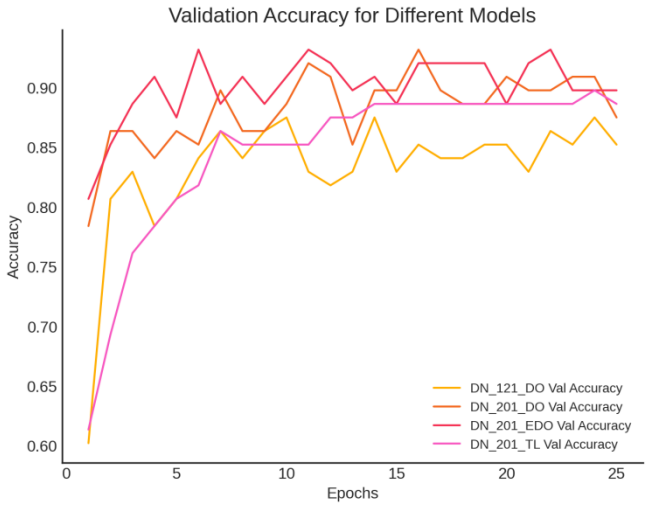

Figure 8. Val Accuracy Over Epochs - Comparison of all Models

When looking at models, ones with less loss on validation and validation that is more accurate are better for use because they show good generalization. Models that settle quickly in loss and accuracy tend to be more efficient. Checking the difference between training metrics and validation metrics is important to find out if there is over-fitting or under-fitting.

When models compared for generalization, rank is Model 1 (DN_201_TL) > Model 3 (DN_121_DO) > Model 4 (DN_201_DO) > Model 2 (DN_201_EDO). Model 1 shows the best mix of training and validating losses, proving strong ability to generalize. Contrastly, Model 2 has bad generalization because of high validation loss and over-fit.

DOI: https://doi.org/10.54216/JCIM.160213
Received: January 06, 2025 Revised: March 05, 2025 Accepted: April 02, 2025

For train loss, order is Model 4 (DenseNet201) < Model 1 (DN_201_TL) < Model 3 (DenseNet121_DO) < Model 2 (DenseNet201_DO). Model 4 gets lowest train loss, showing good learning on training data. However, its variable validation loss makes it less reliable. Meanwhile, Model 2 has highest train loss, meaning slower learning in training.

These rankings show that Model 1 is best-balanced performer, doing well in generalization with stable and low train and validation losses. This comparison is useful for figuring out the best model to use in practical situations.

Table 1: Train Accuracy of all Models

Model	Mean Training Accuracy	Max Training Accuracy	Final Training Accuracy
DN_201_DO	0.92	1	0.99
DN_201_EDO	0.86	0.98	0.97
DN_201_TL	0.82	0.97	0.97
DN_121_DO	0.91	0.99	0.98

D. *Testing Metric Analysis*

DenseNet201 Transfer Learning (DN_201_TL)

A. Engineering Platform:

i. **Correct Classifications**: 20 images correctly classified as "Engineering Platform."

ii. Incorrect Classifications:

a) 3 images incorrectly classified as "Pipeline/Cable."

b) 2 images incorrectly classified as "Underwater Residual Mound."

B. Pipeline/Cable:

i. **Correct Classifications:** 93 images correctly classified as "Pipeline/Cable."

ii. Incorrect Classifications:

a) 2 images incorrectly classified as "Sea Bed Surface."

b) 2 images incorrectly classified as "Underwater Residual Mound."

C. Sea Bed Surface:

i. **Correct Classifications:** 24 images correctly classified as "Sea Bed Surface."

ii. **Incorrect Classifications:**3 images incorrectly classified as "Pipeline/Cable."

D. Underwater Residual Mound:

i. **Correct Classifications:** 38 images correctly classified as "Underwater Residual Mound."

ii. **Incorrect Classifications:** 3 images incorrectly classified as "Pipeline/Cable."

DOI: https://doi.org/10.54216/JCIM.160213
Received: January 06, 2025 Revised: March 05, 2025 Accepted: April 02, 2025

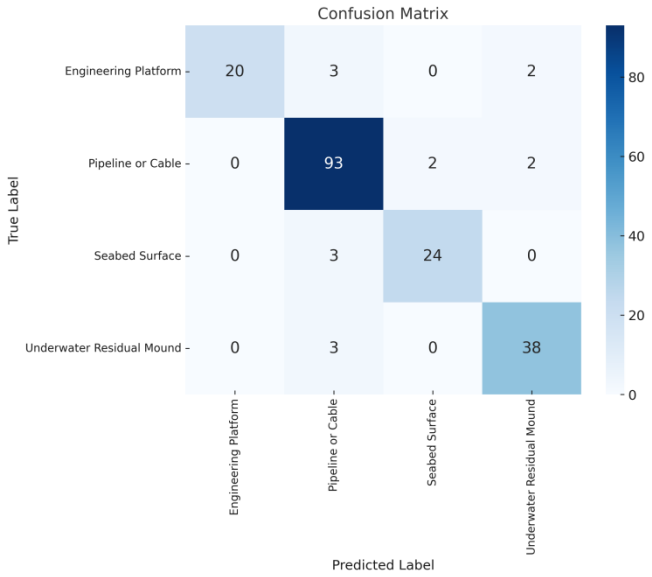

Figure 9. Confusion Matrix - DN_201_TL

DenseNet201 Enhanced Dropout (DN_201_EDO):

A. Engineering Platform:

i. **Correct Classifications**: 24 images correctly classified as "Engineering Platform."

ii. Incorrect Classifications:

a) 1 image incorrectly classified as "Pipeline/Cable."

B. Pipeline/Cable:

i. **Correct Classifications:** 92 images correctly classified as "Pipeline/Cable."

ii. Incorrect Classifications:

a) 2 images incorrectly classified as "Sea Bed Surface."

b) 3 images incorrectly classified as "Underwater Residual Mound."

C. ***Sea Bed Surface:***

i. **Correct Classifications:** 27 images correctly classified as "Sea Bed Surface."

ii. **Incorrect Classifications:** There is no incorrect classification

D. Underwater Residual Mound:

iii. **Correct Classifications:** 39 images correctly classified as "Underwater Residual Mound."

Incorrect Classifications:2 images incorrectly classified as "Pipeline/Cable."

DOI: https://doi.org/10.54216/JCIM.160213
Received: January 06, 2025 Revised: March 05, 2025 Accepted: April 02, 2025

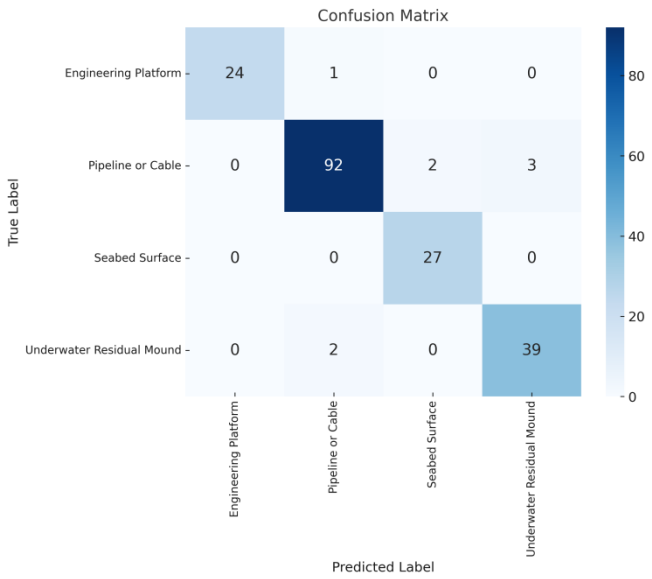

Figure 10. Confusion Matrix - DN_201_EDO

DenseNet121 Dropout (DN_121_DO):

A. Engineering Platform:

i. **Correct Classifications**: 23 images correctly classified as "Engineering Platform."

ii. Incorrect Classifications:

a) 2 images incorrectly classified as "Pipeline/Cable."

B. Pipeline/Cable:

i. **Correct Classifications:** 86 images correctly classified as "Pipeline/Cable."

ii. Incorrect Classifications:

a) 1 image incorrectly classified as "Engineering Platform."

b) 6 images incorrectly classified as "Sea Bed Surface."

c) 4 images incorrectly classified as "Underwater Residual Mound."

C. Sea Bed Surface:

i. **Correct Classifications:** 27 images correctly classified as "Sea Bed Surface."

ii. **Incorrect Classifications:** There is no incorrect classification.

D. Underwater Residual Mound:

i. **Correct Classifications:** 39 images correctly classified as "Underwater Residual Mound."

ii. **Incorrect Classifications:** 2 images incorrectly classified as "Pipeline/Cable."

DenseNet201 Dropout(DN_201_DO):

A. Engineering Platform:

i. **Correct Classifications**: 24 images correctly classified as "Engineering Platform."

ii. Incorrect Classifications:

a) 1 image incorrectly classified as "Pipeline/Cable."

DOI: https://doi.org/10.54216/JCIM.160213
Received: January 06, 2025 Revised: March 05, 2025 Accepted: April 02, 2025

B. Pipeline/Cable:

i. **Correct Classifications:**91 images correctly classified as "Pipeline/Cable."

ii. **Incorrect Classifications:**

a) 3 images incorrectly classified as "Sea Bed Surface."

b) 3 images incorrectly classified as "Underwater Residual Mound."

C. Sea Bed Surface:

i. **Correct Classifications:**27 images correctly classified as "Sea Bed Surface."

ii. **Incorrect Classifications:** There is no incorrect classification

D. Underwater Residual Mound:

i. **Correct Classifications:** 38 images correctly classified as "Underwater Residual Mound."

ii. **Incorrect Classifications:** 3 images incorrectly classified as "Pipeline/Cable."

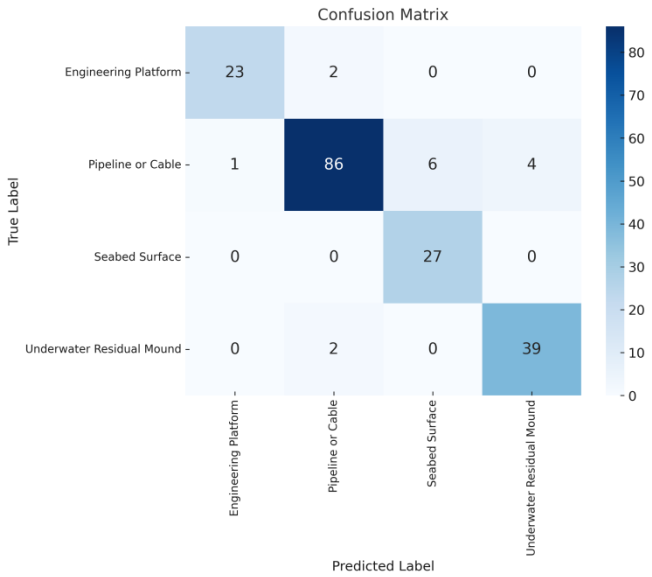

Figure 11. Confusion Matrix - DN_121_DO

All models demonstrate high correct classification rates, ranging from 86.84% to 95.79%, highlighting their effectiveness across all categories. Incorrect classifications remain low, between 4.21% and 13.16%, reinforcing their reliability in maintaining classification accuracy. DenseNet201 Enhanced Dropout (DN_201_EDO) and DenseNet201 Dropout (DN_201_DO) consistently outperform DenseNet121 Dropout (DN_121_DO) and DenseNet201 Transfer Learning (DN_201_TL), exhibiting fewer misclassifications and higher precision. The Pipeline/Cable (P/C) class remains the most challenging for all models, displaying lower correct classification rates and slightly higher error rates compared to other classes. Meanwhile, the Seabed Surface (SBS) and Engineering Platform (EP) classes show superior accuracy with minimal classification errors, reinforcing their robustness.

Overall, the DN_201_DO model performs well by achieving high accuracy throughout while at the same time managing small misclassification rates across categories. The EP class gives 96% accuracy and a 4% misclassification rate—a very reliable category. Likewise, the Underwater Residual Mound (URM) class also reaches the same 96% accuracy, verifying it is model-agnostic. However, P/C classification is harder, with an accuracy of 88% and classifying errors at 12%, leaving more room for improvement. For the SBS category, the value of 94% for accuracy is accompanied by a slightly lower recall of 74%, which may reflect an occasional reluctance to flag at all.

DOI: https://doi.org/10.54216/JCIM.160213
Received: January 06, 2025 Revised: March 05, 2025 Accepted: April 02, 2025

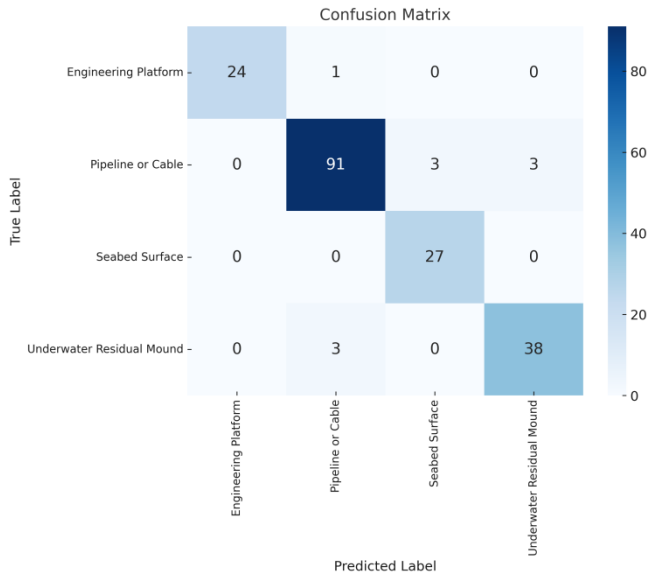

Figure 12. Confusion Matrix -DN_201_DO

DN_201_EDO outperforms the DN_201_DO one, and thus the classification is more robust. Class EP reaches a very high accuracy of 98%, losing only 2% of the misclassification rate, allowing for good reliability. The accuracy for the Pipeline/Cable (P/C) reaches 94% and outperforms the number of misclassifications by DN_201_DO. In addition, even with 2% of rates of misclassification, SBS reaches 98% accuracy. The URM class maintains a high level of accuracy at 96%, indicating that classification is stable.

The strong performance of the DN_201_TL model is, however, consistent throughout but with minor variations in between classes. The EP class achieves 97% accuracy for 3% errors, ensuring reliable classification. However, the correct P/C accuracy of 90% slumps to 92%, and the error P/C rate surges to 8%, indicating a space for improvement. The SBS class is characterized by a very good accuracy of 97% and a low misclassification rate of 3%, which demonstrates the model's strong value for this class.

The DN_121_DO model provides similar performance to the DN_121_DO model and is also a robust classifier. EP remains remarkably accurate with 98% accuracy, and only a 2% misclassification rate attests to its reliability. That was followed by an accuracy of 93% in the P/C class, with precision and recall of 94%, indicating reliable classification. Likewise, SBS achieves an impressive 98%, almost perfect accuracy, and a 2% error rate. This scenario illustrates the robust classification capability of URM separately, where it is stable at 96% accuracy and 4% wrong predictions. Overall, DN_201_EDO appears to have the best P/C misclassification among any model performance in all categories, even with performance improvements from DN_201_TL. The dropout strengthens generalization to a large extent, especially for DenseNet201 derivatives, and it also makes our models more robust for classification tasks in practice.

Table 2: Metrics Comparison of all Models

Metric	DN_121_DO	DN_201_DO	DN_201_EDO	DN_201_TL
Accuracy	0.92	0.95	0.96	0.87
Precision	0.91	0.95	0.96	0.87
Recall (Sensitivity)	0.94	0.96	0.96	0.87

DOI: https://doi.org/10.54216/JCIM.160213
Received: January 06, 2025 Revised: March 05, 2025 Accepted: April 02, 2025

F1 Score	0.92	0.95	0.96	0.87
Specificity	0.97	0.98	0.98	0.96
Negative Predictive Value (NPV)	0.97	0.98	0.98	0.95
Prevalence	1.00	1.00	1.00	1.00
Detection Rate	0.94	0.96	0.96	0.87
Balanced Accuracy	0.96	0.97	0.97	0.91
False Positive Rate (FPR)	0.03	0.02	0.02	0.04
False Negative Rate (FNR)	0.06	0.04	0.04	0.13
False Discovery Rate (FDR)	0.09	0.05	0.04	0.13

The results across the important metrics against the model performance show that DN_201_EDO emerged as the best (with overall 96% accuracy, 96% precision, 96% recall, and 96% F1-score). This is of particular importance since it tells us that the game of DN_201_EDO Leven scored could correctly classify many or most of the identical and switch classes without incurring mistakes in classifying to which class vectors on average actually belong. The follow-up model, DenseNet201 Dropout (DN_201_DO), has slightly worse performance with 95% accuracy but retains high specificity and negative predictive value (both 98%), suggesting its ability for classification.

DenseNet121 Dropout (DN_121_DO) also demonstrates consistent performance with 92% accuracy, 94% recall, and 97% specificity. Although most of its metrics, such as recall, are good, it has slightly worse precision and F1 than DN_201_EDO and DN_201_DO. This means that while its predictions are generally accurate, there is a small compromise between false positives and false negatives. On the other hand, DenseNet201 Transfer Learning (DN_201_TL) has the lowest accuracy (87%), precision (87%), recall (87%), and F1-score (87%) and is the worst performer. The much higher false negative rate (13%) and false discovery rate (13%) indicate that DN_201_TL has difficulty both missing the known positive cases and predicting falsely. Although it has high specificity (96%), it is incapable of optimizing the trade-off between sensitivity and proper classification, resulting in an inadequate overall performance.

A closer look at the detection rate and balanced accuracy shows that DN_201_EDO and DN_201_DO perform better than the other modules, both achieving a 96% detection rate and 97% balanced accuracy. DN_121_DO is a close second, with detection performance remaining high at 94%. However, DN_201_TL lags far behind, as it can only successfully detect at an 87% rate (the best pick, attaining a 91% balanced accuracy), which further reinforces its inadequateness in dealing with varying class distributions.

Error-based metrics also show that DN_201_TL struggles: it has the highest false positive rate (4%) and false negative rate (13%) compared to DN_201_EDO and DN_201_DO, which have much lower error rates (2% FPR and 4% FNR). This means that DN_201_EDO and DN_201_DO do predict more clearly and make fewer mistakes globally.

DOI: https://doi.org/10.54216/JCIM.160213
Received: January 06, 2025 Revised: March 05, 2025 Accepted: April 02, 2025

DN_121_DO works relatively well and has a low false positive rate = 3% and false negative rate = 6%, manifesting that it still possesses good classification stabilization.

Overall, DN_201_EDO emerges as the best-performing model, with strong generalization, high classification accuracy, and minimal errors across all key metrics. DN_201_DO follows closely, showing similar robustness in classification while slightly trailing in recall and precision. DN_121_DO remains a reliable alternative, though it is outperformed by DN_201_EDO and DN_201_DO in most aspects. DN_201_TL, while leveraging pre-trained features, struggles the most in maintaining classification accuracy and reducing false negatives, making it the least reliable model in this evaluation. These findings confirm that dropout-based strategies, particularly in DN_201_EDO, significantly enhance classification performance and generalization, making them the preferred choice for robust model deployment.

6. Conclusion

The study of the Marine_Pulse dataset used DenseNet-DNN structures, which gave clear information on how to classify side-scan sonar images into four marine categories: Engineering Platform (EP), Pipeline/Cable (P/C), Sea Bed Surface (SBS), and Underwater Residual Mound (URM). Different variations of the DenseNet model were implemented for feature extraction and used with DNN Classifier for multi class classification. We determined metrics like accuracy, precision, recall, F1-score, and misclassification rates to see how well the models trained and tested. In all these models, the DenseNet201 Transfer Learning (DN_201_TL model) showed the best ability to perform well on new data, achieving an average training accuracy of 82% (with a highest of 97% and a final of 97%). Its high efficiency over the training metrics indicates that the pre-trained features substantially helped the classifier. Likewise, DenseNet121 Dropout (DN_121_DO) and DenseNet201 Dropout (DN_201_DO) have similar results, where the average training was 91% and 92%, respectively. Both models showed forceful generalization ability, achieving their best training accuracies of 99% and 100% and their best test accuracies of 98% and 99%. The use of dropout techniques was key to reducing overfitting, making those models give more accurate predictions of new data.

Looking at the classes, we obtained the most promising results for the Engineering Platform (EP) and the Seabed Surface (SBS) classes, which had high scores and few misclassifications over all the simulations. In DN_201_TL, the EP class achieved a testing accuracy of 97%, and the SBS class reached a testing accuracy of 97%, indicating that the models can effectively distinguish between the distinctive features of these classes. The Pipeline/Cable (P/C) class, however, was more difficult to classify with testing accuracy of 92% and a higher misclassification rate, demonstrating that P/C's elongated shapes might result in overlaps with other types. Moderate results were obtained with the URM class, where the accuracies from testing were in the range of 94%-96%, and the findings suggested that generalization worked well, but there were specialized tunings to be applied for better performances. Noteworthy, DN_201_EDO showed good generalization performance with an average training accuracy of 86%, a best accuracy of 98%, and a final accuracy of 97%. Even though the training accuracy was a bit, lower than DN_201_DO, its classification remained consistent when tested, which means that the dropout in the model helped it generalize better. Dropout techniques combined with transfer learning were critical to improving model robustness. Dropout reduced overfitting by injecting randomness during training, making models more general. The models exploited transfer learning of the pre-trained weights from the large datasets, which accelerated the learning of the features that were unique to the Marine_Pulse dataset.

This study demonstrates that using the DenseNet framework with DNN as the backbone architecture could effectively solve problems related to side-scan sonar image classification. The Marine_Pulse dataset serves as a solid baseline for marine exploration, underwater structure monitoring, and environmental studying, as it includes various types of natural and human-made underwater objects.

References

[1] A.C. Wölfl, H. Snaith, S. Amirebrahimi, C.W. Devey, B. Dorschel, V. Ferrini, V.A.I. Huvenne, M. Jakobsson, J. Jencks, G. Johnston, et al., "Seafloor Mapping—The Challenge of a Truly Global Ocean Bathymetry," *Frontiers in Marine Science*, vol. 6, p. 283, 2019. doi: 10.3389/fmars.2019.00283.

[2] D. Eleftherakis and R. Vicen-Bueno, "Sensors to Increase the Security of Underwater Communication Cables: A Review of Underwater Monitoring Sensors," *Sensors*, vol. 20, p. 737, 2020. doi: 10.3390/s20030737.

[3] K. Sun, W. Cui, and C. Chen, "Review of Underwater Sensing Technologies and Applications," *Sensors*, vol. 21, p. 7849, 2021. doi: 10.3390/s21237849.

DOI: https://doi.org/10.54216/JCIM.160213
Received: January 06, 2025 Revised: March 05, 2025 Accepted: April 02, 2025

[4] Y. Tang, L. Wang, S. Jin, J. Zhao, C. Huang, and Y. Yu, "AUV-Based Side-Scan Sonar Real-Time Method for Underwater-Target Detection," *Journal of Marine Science and Engineering*, vol. 11, p. 690, 2023. doi: 10.3390/jmse11040690.

[5] R. Kumudham and V. Rajendran, "Super Resolution Enhancement of Underwater Sonar Images," *SN Applied Sciences*, vol. 1, p. 852, 2019. doi: 10.1007/s42452-019-0886-5.

[6] J.J. Schultz, C.A. Healy, K. Parker, and B. Lowers, "Detecting Submerged Objects: The Application of Side Scan Sonar to Forensic Contexts," *Forensic Science International*, vol. 231, pp. 306–316, 2013. doi: 10.1016/j.forsciint.2013.05.032.

[7] N. Nayak, M. Nara, T. Gambin, Z. Wood, and C.M. Clark, "Machine Learning Techniques for AUV Side-Scan Sonar Data Feature Extraction as Applied to Intelligent Search for Underwater Archaeological Sites," in *Field and Service Robotics*, G. Ishigami and K. Yoshida, Eds. Springer, 2021, pp. 1-16. doi: 10.1007/978-981-15-9460-1_16.

[8] C. Langner, W. Knauer, J. Jans, and A. Ebert, "Side Scan Sonar Image Resolution and Automatic Object Detection, Classification and Identification," in *OCEANS 2009-EUROPE*, Bremen, Germany, 2009, pp. 1-8. doi: 10.1109/OCEANSE.2009.5278183.

[9] Y. Dura, Y. Zhang, X. Liao, G.J. Dobeck, and L. Carin, "Active Learning for Detection of Mine-Like Objects in Side-Scan Sonar Imagery," *IEEE Journal of Oceanic Engineering*, vol. 30, no. 2, pp. 360-371, Apr. 2005. doi: 10.1109/JOE.2005.850931.

[10] Kaggle Dataset, "Marine Pulse," [Online]. Available: https://www.kaggle.com/datasets/khajaahmed1/marine-pulse.

[11] C. Li, X. Ye, D. Cao, J. Hou, and H. Yang, "Zero Shot Objects Classification Method of Side Scan Sonar Image Based on Synthesis of Pseudo Samples," *Applied Acoustics*, vol. 173, p. 107691, 2021. doi: 10.1016/j.apacoust.2020.107691.

[12] X. Wang, J. Jiao, J. Yin, W. Zhao, X. Han, and B. Sun, "Underwater Sonar Image Classification Using Adaptive Weights Convolutional Neural Network," *Applied Acoustics*, vol. 146, pp. 145-154, 2019. doi: 10.1016/j.apacoust.2018.11.003.

[13] X. Wang, J. Zhao, B. Zhu, T. Jiang, and T. Qin, "A Side Scan Sonar Image Target Detection Algorithm Based on a Neutrosophic Set and Diffusion Maps," *Remote Sensing*, vol. 10, p. 295, 2018. doi: 10.3390/rs10020295.

[14] X. Ye, C. Li, S. Zhang, P. Yang, and X. Li, "Research on Side-Scan Sonar Image Target Classification Method Based on Transfer Learning," in *OCEANS 2018 MTS/IEEE Charleston*, Charleston, SC, USA, 2018, pp. 1-6. doi: 10.1109/OCEANS.2018.8604691.

[15] K. Sivachandra and R. Kumudham, "A Review: Object Detection and Classification Using Side Scan Sonar Images via Deep Learning Techniques," in *Modern Approaches in Machine Learning and Cognitive Science: A Walkthrough*, V.K. Gunjan et al., Eds. Springer, 2024, pp. 1-20. doi: 10.1007/978-3-031-43009-1_20.

[16] J. Rhinelander, "Feature Extraction and Target Classification of Side-Scan Sonar Images," in *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*, Athens, Greece, 2016, pp. 1-6. doi: 10.1109/SSCI.2016.7850074.

[17] Q. Ge, F. Ruan, B. Qiao, Q. Zhang, X. Zuo, and L. Dang, "Side-Scan Sonar Image Classification Based on Style Transfer and Pre-Trained Convolutional Neural Networks," *Electronics*, vol. 10, p. 1823, 2021. doi: 10.3390/electronics10151823.

[18] B.W. Sheffield, J. Ellen, and B. Whitmore, "On Vision Transformers for Classification Tasks in Side-Scan Sonar Imagery," *arXiv preprint arXiv:2409.12026*, 2024. doi: 10.48550/arXiv.2409.12026.

[19] X. Du, Y. Sun, Y. Song, H. Sun, and L. Yang, "A Comparative Study of Different CNN Models and Transfer Learning Effect for Underwater Object Classification in Side-Scan Sonar Images," *Remote Sensing*, vol. 15, p. 593, 2023. doi: 10.3390/rs15030593.

DOI: https://doi.org/10.54216/JCIM.160213
Received: January 06, 2025 Revised: March 05, 2025 Accepted: April 02, 2025

[20] A. Preciado-Grijalva et al., "Self-Supervised Learning for Sonar Image Classification," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 1-8. doi: 10.48550/arXiv.2204.09323.

[21] U. Anitha and S. Malarkkan, "Analysis of Edge Detection Techniques for Side Scan Sonar Image Using Block Processing and Fuzzy Logic Methods," in *Advances in Intelligent Systems and Computing*, vol. 363, pp. 363–370, 2016. doi: 10.1007/978-981-10-1678-3_35.

[22] G. Huo, Z. Wu, and J. Li, "Underwater Object Classification in Side-Scan Sonar Images Using Deep Transfer Learning and Semi-Synthetic Training Data," *IEEE Access*, vol. 8, pp. 123456-123467, 2020. doi: 10.1109/ACCESS.2020.2978880.

[23] M. Diesing, P. Mitchell, and D. Stephens, "Image-Based Seabed Classification: What Can We Learn from Terrestrial Remote Sensing?" *ICES Journal of Marine Science*, vol. 73, no. 10, pp. 2425–2441, 2016. doi: 10.1093/icesjms/fsw118.

[24] X. Wang, Y. Wang, and M.J. Er, "Review on Deep Learning Techniques for Marine Object Recognition: Architectures and Algorithms," *Control Engineering Practice*, vol. 118, p. 104458, 2022. doi: 10.1016/j.conengprac.2020.104458.

[25] J. Petrich, M.F. Brown, J.L. Pentzer, and J.P. Sustersic, "Side Scan Sonar Based Self-Localization for Small Autonomous Underwater Vehicles," *Ocean Engineering*, vol. 161, pp. 221–226, 2018. doi: 10.1016/j.oceaneng.2018.04.095.

[26] T. Sheffer and H. Guterman, "Geometrical Correction of Side-Scan Sonar Images," in *2018 IEEE International Conference on the Science of Electrical Engineering in Israel (ICSEE)*, Eilat, Israel, 2018, pp. 1-5. doi: 10.1109/ICSEE.2018.8646188.

[27] M. Sung, J. Kim, and M. Lee, "Realistic Sonar Image Simulation Using Deep Learning for Underwater Object Detection," *International Journal of Control, Automation and Systems*, vol. 18, pp. 523–534, 2020. doi: 10.1007/s12555-019-0691-3.

[28] X. Du, Y. Sun, Y. Song, H. Sun, and L. Yang, "A Comparative Study of Different CNN Models and Transfer Learning Effect for Underwater Object Classification in Side-Scan Sonar Images," *Remote Sensing*, vol. 15, no. 3, p. 593, 2023. doi: 10.3390/rs15030593.

[29] M. Einsidler, P. Dhanak, and P.-P. Beaujean, "A Deep Learning Approach to Target Recognition in Side-Scan Sonar Imagery," in *OCEANS 2018 MTS/IEEE Charleston*, Charleston, SC, USA, 2018, pp. 1-4. doi: 10.1109/OCEANS.2018.8604879.

[30] H. Lei and J. Lei, "SI-GAT: Enhancing Side-Scan Sonar Image Classification Based on Graph Structure," *IEEE Sensors Journal*, vol. 24, no. 15, pp. 24388-24404, Aug. 2024. doi: 10.1109/JSEN.2024.3416193.

DOI: https://doi.org/10.54216/JCIM.160213
Received: January 06, 2025 Revised: March 05, 2025 Accepted: April 02, 2025