# Autoencoder-Based Representational Learning for the Determination of Corrosion Severity

**Ikechukwu Ignatius Ayogu[1], Kizito C. Ekebor[2], Euphemia C. Nwokorie[3] and O. C. Nwokonkwo[4]**
[1,3]Department of Computer Science, Federal University of Technology, Owerri, Nigeria
[1]Africa Centre of Excellence in Future Energies and Electrochemical Systems, FUTO
[2]Department of Mechanical Engineering, Federal University of Technology, Owerri, Nigeria
[4]Department of Information Technology, Federal University of Technology, Owerri, Nigeria

**Article Info**

***Article history:***

Received Jul 9, 2023
Revised Oct 26, 2023
Accepted Nov 15, 2023

***Keyword:***

Corrosion severity
Representation learning
Autoencoders
Feature extraction
Corrosion monitoring

**ABSTRACT**

Automatic determination of corrosion severity is an important task that has not received adequate study due to the non-availability of datasets. This study explores corrosion severity detection by leveraging representation learning for the development of lightweight, shallow machine learning models. A variational autoencoder was used for feature extraction. Four classifiers, specifically RF, SVC, k-NN, and XGBoost, were trained using the encoded representation without any additional processing. Voting classifiers were also constructed using the trained models. Except for the slight drop in precision (6.06%) of the soft voting classifier, augmentation produces a general positive influence on the precision, recall, and F1-score of the models understudied; it improved the precision, recall, and F1-metrics of K-NN and hard-voting classifiers most remarkably. For K-NN, the improvement reached 29.17%, 20.93%, and 30.77% for precision, recall, and F1-metrics, respectively, and 30.36%, 26.42%, and 35.29%, respectively, for the hard voting classifier. Whereas our experiments could not produce state-of-the art results, it provides adequate motivation to further study VAE as a data preprocessing unit for the development of simple, efficient, lightweight models that can be deployed on resource constrained devices, which will in turn advance the development and deployment of corrosion monitoring systems on low-cost devices.

***Corresponding Author:***

Ikechukwu Ignatius Ayogu,
Department of Computer Science,
Federal University of Technology,
Owerri, Nigeria
Email: Ignatius.ayogu@futo.edu.ng

## 1. INTRODUCTION

Corrosion is a chemical process that gradually converts an unstable metal to its more stable oxide through electrochemical processes [1–4]. This process incrementally reduces the strength of the metallic structure, leading to structural weakness and eventual loss of structural integrity. The resultant effect is an eventual facility failure that brings about unpleasant consequences. Chemists tend to agree that it is technically impossible to eliminate corrosion since non-precious metals are chemically unstable by nature [5–7]. Therefore, increased positive outcomes from corrosion mitigation actions [8–9] are contingent upon the timely detection of corrosion. However, manual inspection to detect the onset or existence of corrosion in facilities is characterized by a number of significant challenges. It is time-consuming, expensive, requires experienced personnel, and most importantly, it is often unable to detect the onset of corrosion, especially in parts of a facility that are difficult to access [10–11].

Corrosion prevention and management involve a carefully executed plan to detect the onset of corrosion and apply appropriate remediation actions to slow down the rate of corrosion, either through the use of specialized coatings or inhibitors. Corrosion monitoring by visual inspection is a challenging process; it is time-consuming, expensive, and risky to conduct in large facilities [12–13]. The challenges associated with corrosion inspection and monitoring, according to research, are due to accessibility issues, diversity of corrosion types, the somewhat slow but steady progression of the corrosion, which makes monitoring and accurate prediction of corrosion severity difficult, the inability of detection methods to detect the onset of corrosion, the cost of implementing a real-time corrosion monitoring program, the cost of inspection, a lack of good data interpretation techniques, environmental factors, and inefficiencies and inadequacies in the corrosion inspection and monitoring techniques [12–14].

Among these challenges, the accessibility issue with corrosion sites is very significant. Corrosion can occur in hard-to-reach areas or hidden locations, which makes visual inspection or monitoring the extent of corrosion extremely difficult. Current inspection techniques cannot efficiently assess corrosion incidence in parts of facilities that are underground, such as buried pipelines or the internal surfaces of the pipelines [15–18].

The use of computer vision for corrosion detection offers a promising solution to automate the detection of corrosion through the development of intelligent, self-guided inspection tools, and with the increasing availability of datasets, the future of corrosion informatics is promising. Image processing, machine learning, 3D imaging, and drones equipped with cameras represent the most popular techniques for corrosion monitoring and detection. However, the use of computer vision for corrosion detection is still in its early stages of application, and therefore, the need for further research and development to improve the accuracy and reliability of these techniques exists [18, 19]. Research has explored different types of computer vision techniques for corrosion detection. The most common technique has been image processing, which analyzes images of corroded surfaces to detect corrosion. Strings of research have explored a range of these techniques, but the most popular is the use of texture analysis techniques for thresholding, segmentation, and feature extraction. Some representative studies are [17], [20], and [21].The use of computer vision for corrosion detection offers a promising solution to automate the detection of corrosion through the development of intelligent, self-guided inspection tools, and with the increasing availability of datasets, the future of corrosion informatics is promising. Image processing, machine learning, 3D imaging, and drones equipped with cameras represent the most popular techniques for corrosion monitoring and detection. However, the use of computer vision for corrosion detection is still in its early stages of application, and therefore, the need for further research and development to improve the accuracy and reliability of these techniques exists [18, 19]. Research has explored different types of computer vision techniques for corrosion detection. The most common technique has been image processing, which analyzes images of corroded surfaces to detect corrosion. Strings of research have explored a range of these techniques, but the most popular is the use of texture analysis techniques for thresholding, segmentation, and feature extraction. Some representative studies are [17], [20], and [21].

The importance or relevance of features derived from image data to the target concept is known to positively influence classification outcomes [22–26]. Features that are highly relevant to the target concept enhance an algorithm's capacity to differentiate between spectrally similar classes, leading to improved classification performance. Although commonly used linear feature extraction methods like principal component analysis (PCA) and linear discriminant analysis (LDA) are simple and easily implemented, they are unable to efficiently model the nonlinear structures in the data. The exceptional state-of-the-art performance of deep learning models is primarily attributed to their ability to learn non-linear features from data. However, the computational intensity of deep learning modeling and deployment makes it challenging to implement on resource-light devices [27, 28]. Given the growing ubiquity of resource-light devices, there is a desire for corrosion monitoring tools that are directly executed on low-power, resource-light devices. This paper investigates the use of neural network-based feature extraction techniques to extract relevant features to train shallow machine learning models for corrosion severity classification. The aim is to leverage the superior feature learning capabilities of deep learning models without incurring computational overhead during runtime.

Automatic determination of corrosion severity is an important task that has not received adequate study due to the non-availability of datasets [18]. This paper utilized the only published corrosion severity classification dataset described in [18]. However, in recent times, corrosion informatics research has continued to benefit from breakthroughs in deep learning. Many researchers have proposed different deep learning models for corrosion monitoring and detection in a variety of applications, such as corrosion detection in aircraft fuselage, as pioneered by [29–31] and recently by [14], who proposed a deep learning-based corrosion detection in aircraft fuselage. [32] sought to improve the performance of deep learning models for corrosion detection in civil structures by using semantic segmentation instead of pixel segmentation approaches. While the combination of two convolutional neural network models produced improved performance, the authors failed to assess the implications of stacking these two models on compute requirements. [33] successfully combined the use of drones and deep learning approaches for the detection of corrosion in large industrial buildings using drone-acquired images with a view to reducing the challenges of corrosion inspection in such facilities. Others are the application of Bayesian deep learning for corrosion detection [34] and a comparative study of the standard computer vision techniques for corrosion detection.

The deep learning models produce top performance in many tasks, including computer vision applications. However, unlike shallow learning algorithms, they are notable for their high computational requirements, which limit their deployability on resource-constrained devices such as low-end mobile phones and edge devices [35]. The superior performance delivered by deep learning models is linked to their ability to learn features in situ, unlike the shallow learning algorithms that are trained on hand-engineered features. This paper studied the effect of integrating representation learning and shallow models for corrosion severity rating. Experiments were conducted to investigate how the use of the learned representation affected the ability of Random Forests (RF), Support Vector Machines (SVM), Extreme Gradient Boost (XGBoost), and k-NN algorithms on the corrosion severity classification task. In particular, we trained, evaluated, and fine-tuned a variational autoencoder for representation learning. Once a satisfactory limit was achieved, the decoder was separated, and the bottleneck was linked to the shallow learners to serve as a source of features. The anticipation is that the representation learning unit based on deep learning will alleviate the inefficiencies associated with manually crafted features and provide performance advantages to shallow learners without escalating their computational demands.

## 2. APPLICATIONS OF GENERATING CAPTION FOR AERIAL IMAGES

### 2.1. Dataset and Data Pre-processing

This dataset used for this research was created by [18]. It features 600 images of corroded 4 by 6-inch metal panels. The images are labeled with expert-validated corrosion severity scores ranging from 5 to 9, with 5 being the most severe. Images in the dataset contain either a single or double scribe across the surface of the metal, featuring different background colors for the same rating as well as the likelihood of noise around the actual corrosion. According to [18], the ratings of the panels in the dataset were accessed in the range of 5–9 because a rating lower than 5 is considered a failed test in the corrosion rating domain. The data set is balanced, featuring 120 images in each of the four rating classes.

For all the experiments reported in this paper, the images were resized to 256 x 256 x 3 pixels, and pixel values were normalized between 0 and 1. The effects of some manual image augmentation, namely, rotation, random zooming, random flipping, random width and height shifting, and random shearing, on the performance of the VAE model were experimented with. The maximum allowed fraction for zooming, flipping, shifting, and shearing was 3%, while 3 degrees was used for rotation. According to the literature [50, 51], this range ensures that the images from augmentation are like the original images. Samples from the augmented images are shown in Figure 1. Features extracted from the encoder were centered to a zero mean and a standard deviation of 1 before feeding them into the downstream classifiers. The VAE was used as a data preprocessing tool in this research.

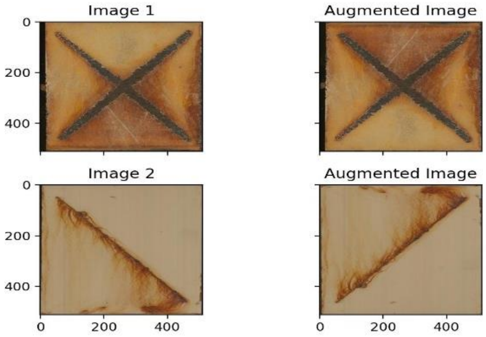

Figure 1. Sample images and their augmented versions created using the Combined+ augmentation techniques.

### 2.2. Methods

#### 2.2.1. Variational Autoencoders, VAEs

An autoencoder is a form of neural network (mostly RNNs and CNNs) where the input and output are identical. As depicted in Figure 2, an autoencoder has three components, namely: an encoder, the code, and a decoder.

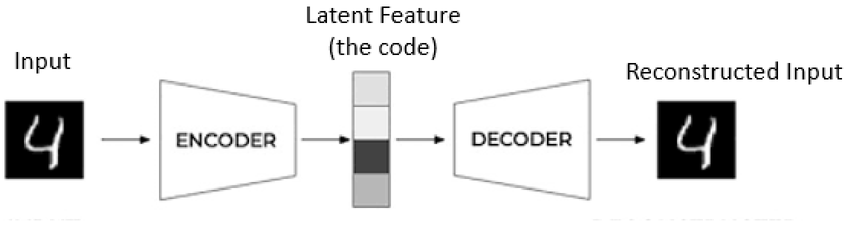

Figure 2. The components of an autoencoder

The code is produced by the encoder through an unsupervised compression of the input; the decoder takes the code and reconstructs the input. Autoencoders are trained via back propagation [36]. Autoencoders are lossy, and the goal, therefore, is to train the model to minimize the reconstruction error. One option for network structure is to make the network's core smaller (represented by fewer numbers) than the input. Without labels, an autoencoder learns a condensed representation of the input. As a result, it is an unsupervised learning technique that may be used on enormous amounts of unprocessed data without requiring laborious annotation procedures. In a landmark study on autoencoders, [37] demonstrated that, compared to principal components analysis [38], autoencoders produced compact representations of images that are more accurate representations for reconstructing the original images.

The ability of autoencoders to be modified to learn various patterns, irrespective of the task, is one of their most significant advantages. An example of this adaptation can be found in [39], where a recurrent neural network was trained to learn the temporal relationship in audio files and subsequently used to produce noise-free audio from noisy audio. [40] and [41] have employed VAE for feature extraction for classification in different scenarios. VAEs are a more recent alternative to RNNs and CNNs for adjusting neural networks to capture salient features in data [42, 43]. Autoencoders, in general, are frequently trained to reduce reconstruction loss—the difference between the reconstructed inputs and the actual inputs [26]. Although the resulting representation may be exceedingly accurate and precise, it may also have unexpected distributions that make it unsuitable for feeding into supervised classifiers [44]. VAEs address this problem by minimizing two objective functions: the reconstruction loss and the distribution of values, and a difference between the distributions of values in the representation to a Gaussian distribution of zero mean and unit variance [42, 43, 26].

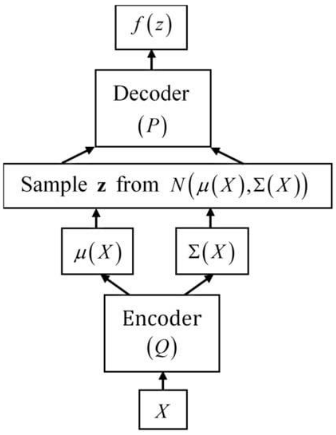

Figure 3. A realization of VAE architecture [46]

VAEs (Figure 3) are generative models that share the structural components of the autoencoders but differ mathematically in a significant proportion [47]. A VAE learns a latent variable model of the input data by constraining the encoded representation. Given a vector of sample $\boldsymbol{x} \in \mathbb{R}^{\boldsymbol{n}}$, the encoder part of the VAE defines a Gaussian probability distribution in the latent space [42]. Practically, this is represented as an additional variational layer that is composed of a mean vector $\mu \in \mathbb{R}^{m}$ and a standard deviation vector $\boldsymbol{\sigma} \in \mathbb{R}^{m}$ with the same dimension as the latent vector. The decoder draws samples from the latent distribution represented by the vector $\boldsymbol{z} \in$ to generate a new data point [46].

To successfully train a VAE model, solutions to a two-part optimization problem are required: (i) reducing the MSE and (ii) KL-divergence losses [48]. Solving these problems is challenging. It is difficult to determine the optimum weight of the distribution component that is needed to ensure that the model reconstructs the input accurately. The weight of the distribution component is commonly constrained to its optimum by first optimizing the reconstruction of the inputs, followed by a gradual imposition of Gaussian restriction on the bottleneck [49]. The VAEs are composed of a probabilistic encoder Q, $\boldsymbol{q\theta}(\boldsymbol{z} \mid \boldsymbol{x})$, that tries to approximate the true posterior distribution $\boldsymbol{p}(\boldsymbol{z}|\boldsymbol{x})$, and a generative decoder P, $\boldsymbol{p\phi}(\boldsymbol{x} \mid \boldsymbol{z})$, that reconstructs the input by sampling from the latent representation described in the code, independent of the input, $\boldsymbol{x}$ [47].

### 2.2.2. Model Architecture

The overall framework for the investigations reported in this paper consists of two parts: a variational autoencoder model (Figure 4) for representation learning and a system of shallow classifiers (Figure 5) that are used to predict corrosion ratings based on the informative features extracted from the code component of the variational autoencoder.

Our experimental VAE architecture consists of five convolutional blocks and a final latent block in the encoder, and an initial latent block and five deconvolutional blocks in the decoder. Each convolutional block comprises a convolutional layer to extract relevant features, a batch normalization layer for normalization and regularization, and a leaky rectified linear unit (ReLU) layer that introduces nonlinearity to the model and enables it to learn complex patterns in the data.

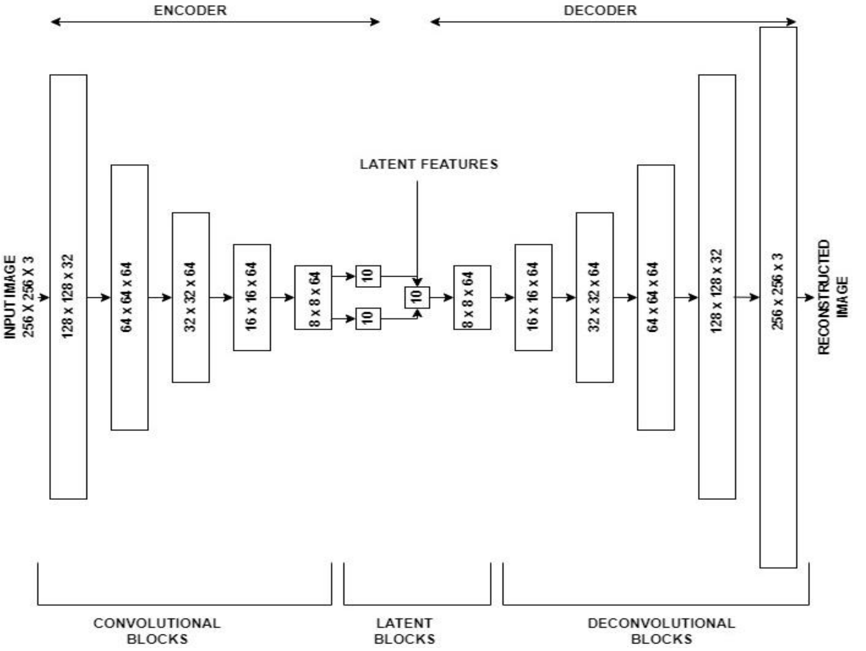

Figure 4. The architectural details of the experimental VAE model

The final latent block takes output from the convolutional blocks and generates two ten-dimensional vectors that represent the distributions of the ten-dimensional feature space used for classifier training. Similarly, the decoder has an initial latent block and five deconvolutional blocks, each having a transpose convolutional layer, a batch normalization layer, and a leaky rectified linear unit layer. The decoder mirrors the encoder and upscales the latent features back to the original image size. To identify an optimal configuration tailored for our specific task, the selection of the VAE architecture, including choosing a dimension for the latent space, filter sizes, and kernel sizes, was made through iterative experimentation and parameter search using Bayesian optimization (Table 2). Table 2 also contains the optimal parameters of the classifiers that were investigated.

To leverage the strengths and diversity of multiple classifiers and improve the overall prediction performance, this paper considered voting classifiers as a fusion technique. The predictions by each classifier were combined through voting. Experiments were carried out to investigate the efficiency of hard and soft voting techniques. The final prediction was made using the hard and soft voting approaches. In the case of hard voting, each classifier casts a vote, and the class with the most votes is considered the final prediction, while in the soft voting setup, each classifier provides probability scores for each of the classes, and the class with the highest average probability score is chosen as the final prediction. Our experiments explored non-weighted soft and hard voting techniques.

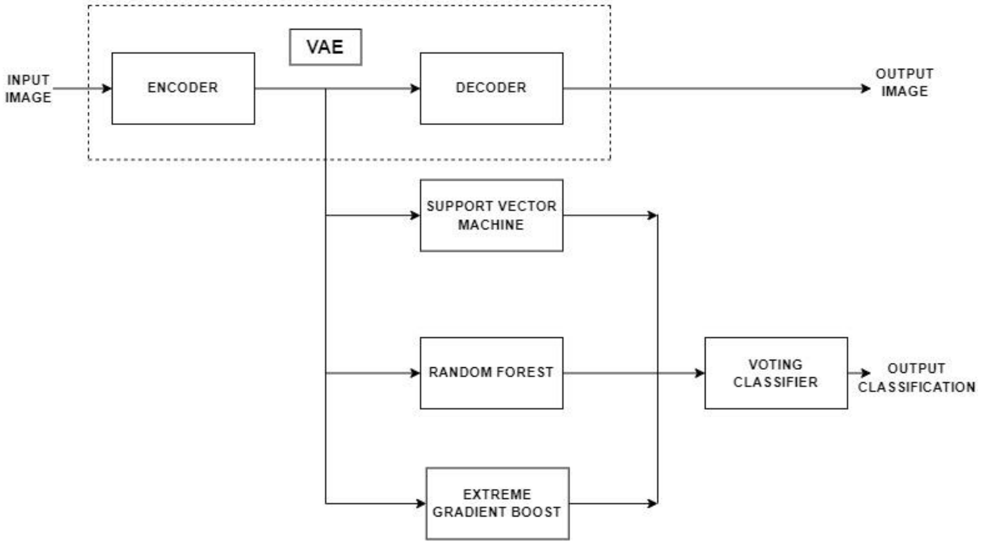

Figure 5. Experimental setup

## 3. EXPERIMENTS

Our experiments were designed to explore the prospects of combining representation learning and shallow machine learning algorithms for corrosion severity prediction and evaluate the performance of a voting classier ensemble that aggregates the contributions of each of the best classifiers in the pool. The overall motivation is to seek ways to develop a light-weight model that can significantly match the performance of state-of-the-art but compute-heavy deep learning models.

### 3.1. Model Training Setup

(a) VAE Training

The training involved optimizing the network parameters to learn an efficient latent representation for corrosion severity classification. The VAE network parameters were optimized to minimize the mean squared error (MSE) loss and the Kullback-Leibler (KL) divergence loss. The MSE loss measures, on a pixel-by-pixel basis, the mean square difference between the original image fed into the encoder and the output image from the decoder, while the KL divergence loss measures the difference between the distributions from the latent feature space and the standard normal distribution. This helps to ensure that the latent representations are meaningful. A scaling factor, beta [43], was used to ensure the right balance between both loss functions. The total loss is the sum of the reconstruction loss, scaled by beta, and the KL loss. The model parameters were optimized using the Adams optimizer with an exponentially decaying learning rate of 0.001. The optimal hyper-parameter values used to train the VAE for this task are as shown in Table 1.

Table 1. Optimal hyper-parameter values for VAE training

| Hyper-parameter | Search Space | Optimal hyperparameter |
|---|---|---|
| Beta | 100 – 10,000 | 660 |
| Latent Dimension | 2-250 | 10 |
| Initial Learning Rate | 0.0005 – 0.05 | 0.001 |
| Decay Steps | 200 - 600 | 400 |
| Decay Rate | 0.95 - 0.5 | 0.7 |
| Input Image Size | 28 - 512 | 256 |

(b) Downstream Classifier Training

The learned representations from the Variational Autoencoder were used to train four (4) downstream classifiers, namely, a random forest classifier (RF), a support vector classifier (SVC), a k-nearest neighbor classifier (k-NN), an extreme gradient boost classifier (XGBoost), and a voting classifier that exploits the strengths of the three best-performing classifiers. The optimal parameters for each of these models, shown in Table downstream, were obtained through the Bayesian optimization search technique [53]. To further explore the avenues for improved performance, a voting classifier whose output is an aggregate of its component estimators was trained using the soft voting and hard voting aggregation techniques, respectively. While the output of the soft voting technique is based on the cumulation of the predicted probabilities from the component estimators, the hard voting technique takes the absolute majority of the classifications from the component estimators.

Table 2. Optimal hyperparameters for the classifiers

| Classifier | Hyper-parameter | Search Space | Optimal hyperparameter |
|---|---|---|---|
| RF | Number of Estimators | 50 – 500 | 351 |
| | Max Depth of Estimator | 6 - 10 | 6 |
| | Bootstrapping | (True, False) | True |
| | Features per subsample | (log$_2$(10), 10, sqrt (10)) | Sqrt (10) |
| | Min Leaf Sample | 2 - 8 | 2 |
| | Min Samples for Split | 2 - 8 | 5 |
| SVM | Regularizer | 0.01 - 2 | 1.547 |
| | Kernel | (Linear, RBF) | RBF |
| XGBoost | Number of Estimators | 25 - 800 | 50 |
| | Max Depth of Estimator | 3 - 10 | 4 |
| k-NN | Number of Neighbors | 5 - 45 | 39 |

## 4. RESULTS AND DISCUSSION

### 4.1. Evaluation of the Variational Autoencoder Model

Our VAE model was evaluated based on MSE and KL divergence losses. These losses simply measure the difference between the input image and the reconstructed image. Table 3 presents the overall effect of data augmentation on the VAE model. As shown in Table 3, the VAE model trained without augmentation achieved a training loss of 9.17 and a mean squared error (MSE) of 0.0108 and 0.0107, respectively, on the training and validation data. Our experiments investigated the effect of flipping, rotation, and a combination of random shear, random zoom, width, and height shift (named Combined+) on the performance of the VAE model, and the results show that these exert a negative effect. As Table 3 shows, horizontal flip (Hflip) increases the training loss by 2.03 points, vertical flip (Vflip) by 0.89 points, and rotation by 1.99 points, while combined+ increases the training loss by 5.4 points. It is remarkable that these poorer indices, which indicate increased model perplexity, did not deteriorate the performance of the downstream classifiers per se. Also, the validation losses for each of the augmentation techniques clearly show that the augmentation techniques did not impair the performance in any appreciable degree. This is evident in Figure 6.

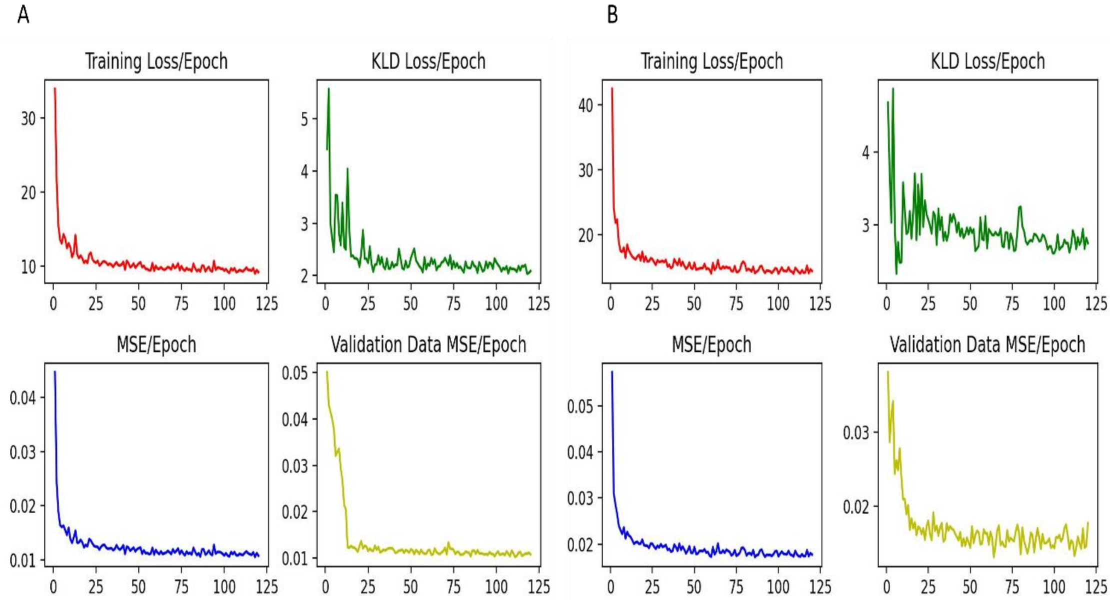

Figure 6. Plot of losses against epochs (A) without augmentation; (B) without augmentation.

Table 3. Effect of augmentation on training and validation losses

| Augmentation | Training loss | Validation loss (MSE) |
|---|---|---|
| None | 9.17 | 0.0108 |
| Hflip | 11.2 | 0.013 |
| Vflip | 10.06 | 0.011 |
| Rotation | 11.16 | 0.011 |
| Combined+ | 14.57 | 0.017 |

The MSE for the model without augmentation also supports the fact that the model did not overfit, even though there is a marginal increase in validation losses as shown in Table 3. A cross-section of the images generated by the VAE models is shown in Figure 7.

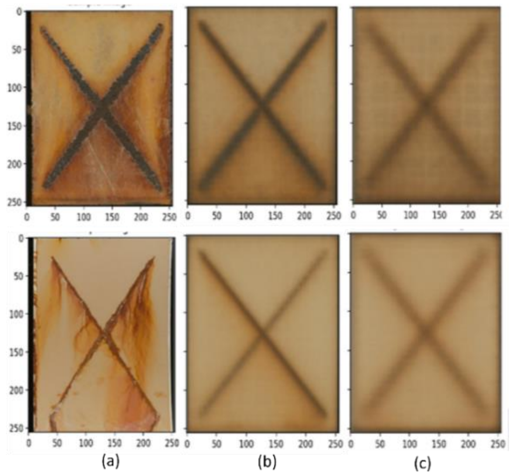

Figure 7. Samples of images generated by an untrained decoder, a decoder trained with image augmentation, and a decoder trained without image augmentation

### 4.2. Evaluation of Classifiers Using VAE Learned Features

Our experiments investigated the effects of VAE-learned features on the performance of the models under study. The precision, recall, and f1-score metrics of the models presented in Table 4 and visualized in Figure 6 (a), (b), and (c), respectively. These scores were recorded for two groups of VAE: VAE-1 and VAE-2, representing VAE that is trained with image augmentation and without augmentation, respectively.

Except for the slight drop in the precision (6.06%) of the soft voting classifier, the trend in Table 4 indicates that the VAE model that is trained with image augmentation produced a general positive influence on the precision, recall, and F1-score of the models under study. However, these positive effects on the performance of the models have a degree of variability across the models. It improves the precision, recall, and f1-metrics of K-NN and hard-voting classifiers most remarkably. For K-NN, the improvement reached 29.17%, 20.93%, and 30.77% for precision, recall, and f1-metrics, respectively, and 30.36%, 26.42%, and 35.29%, respectively, for the hard voting classifier. As the trend in Figure 6 shows, the use of data augmentation techniques generated better VAE models since the features learned from these categories of VAEs produced a positive improvement for recall and F1-score across all the models. This is in line with the literature.

Table 4. Effects of combined augmentation on the quality of features produced by the VAE

| Model | Precision | | Recall | | $F_1$-score | |
|---|---|---|---|---|---|---|
| | VAE-1<br>Combined+ | VAE-2<br>Non-Augmented | VAE-1<br>Combined+ | VAE-2<br>Non-Augmented | VAE-1<br>Combined+ | VAE-2<br>Non-Augmented |
| k-NN | **0.62** | 0.48 | **0.52** | 0.43 | **0.51** | 0.39 |
| RF | 0.65 | 0.58 | 0.63 | 0.55 | 0.61 | 0.54 |
| XGBoost | 0.56 | 0.54 | 0.55 | 0.52 | 0.55 | 0.51 |
| SVM | 0.64 | 0.61 | 0.58 | 0.52 | 0.59 | 0.52 |
| V-soft | 0.62 | **0.66** | 0.62 | 0.58 | 0.61 | 0.59 |
| V-hard | **0.73** | 0.56 | **0.67** | 0.53 | **0.69** | 0.51 |

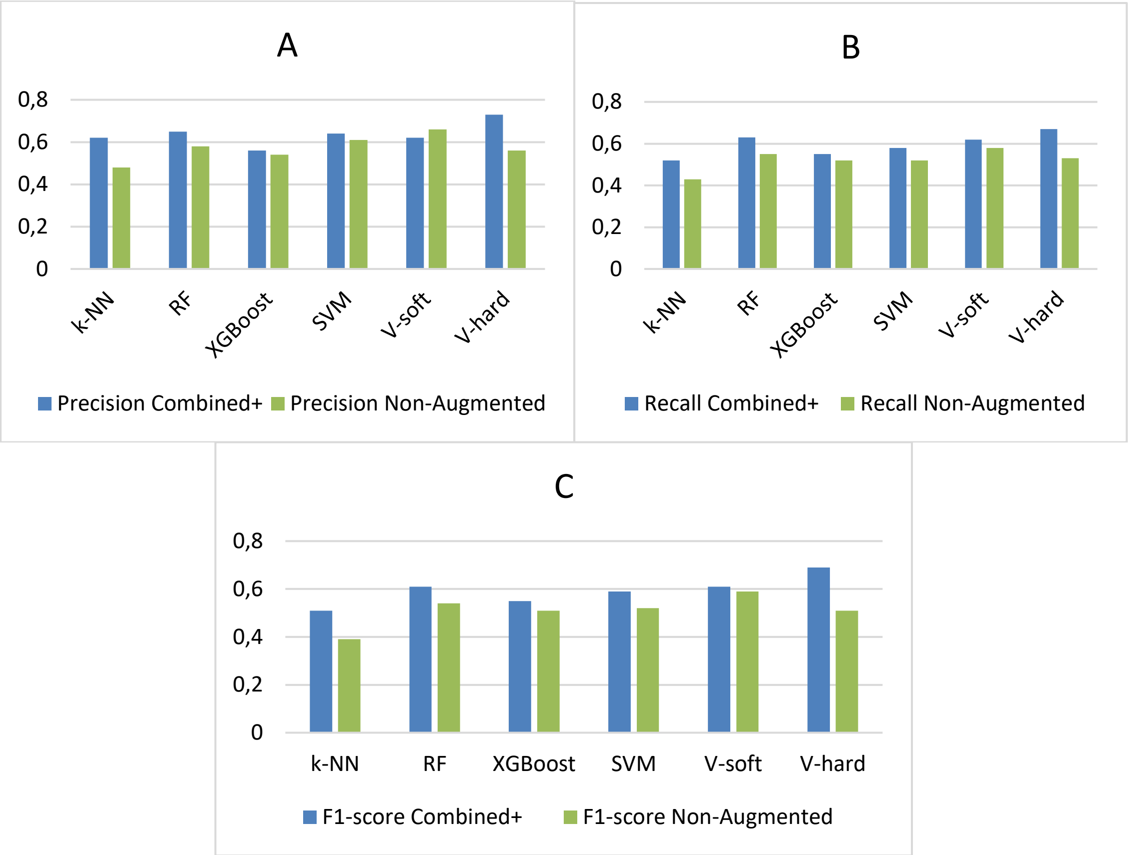

Figure 8. Effect of augmentation on the precision (a), recall (b) and $f_1$-score (c) of the classification models.

We studied how horizontal flipping, vertical flipping, rotation, and a mix of these techniques, along with a combination of them, affected the accuracy of the downstream classifiers on corrosion images that had not been seen before. The results are shown in Table 5. As can be observed, rotation enabled the VAE model to generate features that caused the RF, SVM, KNN, and Hard voting classifiers to attain accuracy of 67%, 63%, 62%, and 67%, respectively, representing an improvement of 14, 11, 19, and 14 points, respectively. Thus, out of the six models investigated, the VAE learned on rotation-augmented data produced the best improvement on the classifiers, and thus, it indicates that rotation influences the generation of the most relevant features by the VAE, which in turn translates to better model performance downstream. Also, horizontal flipping (Hflip) had the same effect on KNN's performance as rotation and made the most progress on the soft voting classifier, achieving 67% accuracy, which is 9 points higher than the starting point. It is also observed that vertical flipping, Vflip, has as much influence on the ability of the VAE to improve the accuracy of the Hard Voting Classifier as rotation.

It is remarkable that although none of the classifiers understudied could attain the accuracy reported in [18], the recorded accuracy scores offer some promise that motivates further investigation of this framework. While [18] reported higher accuracy scores than we have obtained, evidence from the literature suggests that the performance of the deep learning models ResNet-18, ResNet-50, DenseNet and HRNet on this dataset in [18] is much lower than the state-of-the-art for the deep learning models under reference. For instance, ResNet already achieves over 95% for image classification using ImageNet dataset. This therefore suggests that the dataset is difficult.

Table 5. Effect of various VAE variants on the accuracy of the classifiers

| VAE Type | Test Accuracy (single models) | | | | Test Accuracy<br>(voting classifiers) | |
|---|---|---|---|---|---|---|
| | RF | SVM | KNN | XGB | Soft | Hard |
| VAE + NA | 55 | 52 | 43 | 52 | 58 | 53 |
| VAE + Hflip | 62 | 62 | **62** | 63 | **67** | 65 |
| VAE + Vflip | 65 | 60 | 58 | **65** | 65 | **67** |
| VAE + Rotation | **67** | **63** | **62** | 55 | 63 | **67** |
| VAE + Combined+ | 63 | 58 | 52 | 55 | 60 | 65 |

To ascertain performance on completely unseen data, the best performing classifier, the hard voting classifier, was subjected to a completely unseen set of images. This was performed twice, first using feature vectors from the VAE trained without image augmentation and then using feature vectors from the VAE trained with image augmentation. In the former, the classifier performed relatively poorly, with an accuracy of 53% in comparison with the 69% accuracy obtained with the validation set. In the second case, the classifier achieved an accuracy of 65% on the test set. This suggests that while image augmentation seemed to hurt the efficiency of the features generated by it, it reduced the variance between the training set and the test set, as the confusion matrix in Figure 9 indicates. While there are impressive results, especially from classes 9 and 8, this confusion matrix shows that our architecture has some difficulty distinguishing two neighboring corrosion severity classes and the most difficulty spotting class 6. The reason for this observed difficulty is attributable to inadequacies in the VAE's ability to encode the images, as the cluster diagram in Figure 10 shows.

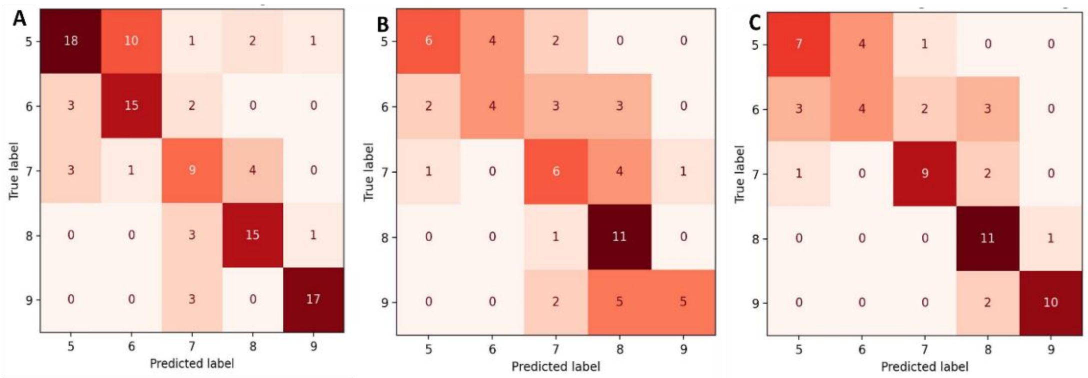

Figure 9. Confusion matrix from best classifier (A) using feature vectors from validation set, (B) using feature vectors of test set from VAE trained without image augmentation, and (C) using feature vectors of test set from VAE trained with image augmentation.

The feature vectors or encodings from the trained encoder of the VAE were clustered after shrinking them to principal components. This visualization showed some degree of clustering, but a strong bias is noticeable. A further reduction in bias is expected to produce better separation, which will in turn produce better generalization ability. Figure 10 reveals some overlaps between the clusters of feature vectors from neighboring classes, and this points to the fact that, in addition to the quality and size of the dataset, these observed overlaps contribute to the misclassifications. Although this reveals weaknesses in the encoder architecture, it is a promising approach towards achieving a lightweight encoder-classifier model for corrosion severity classification.

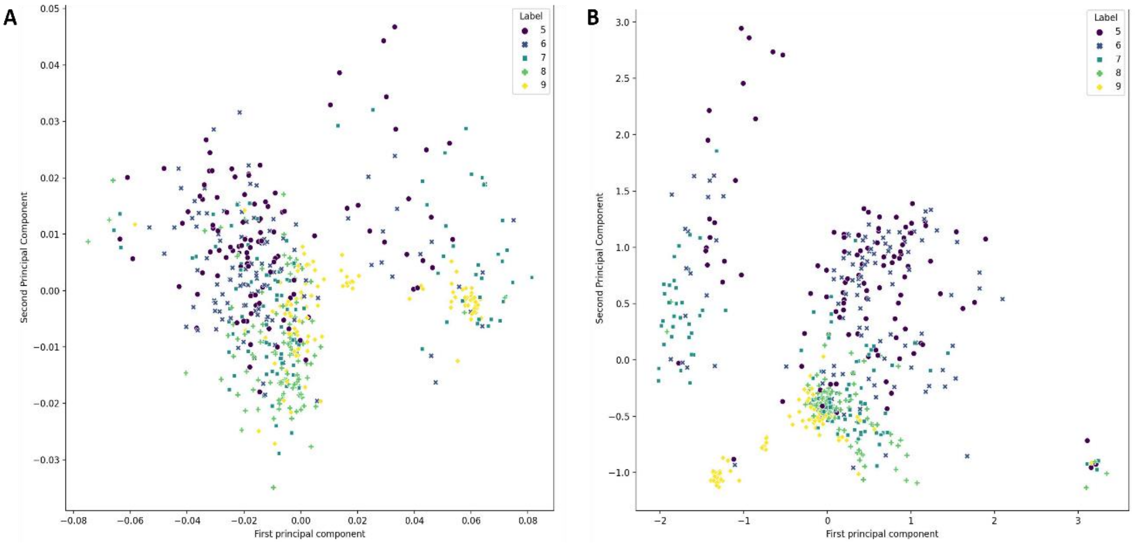

Figure 10. 2 D visualization of image encodings before (A) and after training the VAE (B). No clusters are seen in A while B shows some level of clustering.

## 5. CONCLUSION

We investigated an encoder-based representational learning setup for corrosion severity classification using a recently published dataset. We setup a variational autoencoder as a feature extraction unit and train four classifiers, namely, RF, SVC, k-NN, and XGBoost, off the encoded representation without further treatment. We built a voting classifier using the trained RF, SVM, and XGBoost models, which were the three best-performing models. We also looked into how rotation and flipping affected the quality of the features produced by the variational autoencoder by measuring their influence on downstream classification algorithms. Results reveal that a VAE that implements rotation and vertical flip produced features that produced equal improvement in the performance of the hard voting classifier, while a VAE that implements horizontal flipping generated features with the highest relevance to the performance of the soft voting classifier. Overall, our results indicate that this scheme is promising, but further exploration is required to achieve the state-of-the-art. Particularly, we noticed that the VAE architecture needs further refinement to further increase its feature-learning competence. This will be the focus of our future work.

## REFERENCES

[1] Shreir, L. L. Basic Concepts of Corrosion. Shreir's Corrosion. Oxford: Elsevier, (2010), 89-100.

[2] Cicek, V. and Al-Numan B. Corrosion chemistry. John Wiley & Sons, (2011).

[3] Mattsson, E. Basic corrosion technology for scientists and engineers. CRC Press, 2023.

[4] Daoudi, W. El Aatiaoui, A. Echihi, S. El Ibrahimi, B. and Oussaid, A. Corrosion inhibition: current trends and challenges. Computational Modelling and Simulations for Designing of Corrosion Inhibitors, (2023), 3-21.

[5] Davis, J. R. The effects and economic impact of corrosion. In Corrosion: Understanding the basics, Materials Park: ASM International, 2000, Vol. 164, pp. 62-66.

[6] Oguzie, E. E. Corrosion inhibition of aluminium in acidic and alkaline media by Sansevieria trifasciata extract. Corrosion science, (2007), 49(3), 1527-1539.

[7] Fayomi, O. S. I. Akande, I. G. and Odigie, S. Economic impact of corrosion in oil sectors and prevention: an overview. In Journal of Physics: Conference Series, 2019, December, Vol. 1378, No. 2, p. 022037. IOP Publishing.

[8] Coelho, L. C. C. dos Santos, P. S. S. da Silva Jorge, P. A. Santos, J. L. and de Almeida, J. M. M. M. Real-time early warning strategies for corrosion mitigation in harsh environments. Journal of Lightwave Technology, (2018), 36(4), 1152-1158.

[9] Thakur, A. and Kumar, A. Sustainable inhibitors for corrosion mitigation in aggressive corrosive media: a comprehensive study. Journal of Bio-and Tribo-Corrosion, (2021), 7, 1-48.

[10] Popoola, L. T. Grema, A. S. Latinwo, G. K. Gutti, B. and Balogun, A. S. Corrosion problems during oil and gas production and its mitigation. International Journal of Industrial Chemistry, (2013), 4, 1-15.

[11] Wu, R. Zhang, H. Yang, R. Chen, W. and Chen, G. Nondestructive testing for corrosion evaluation of metal under coating. Journal of Sensors, (2021), 1-16.

[12] Yu, L. Yang, E. Luo, C. and Ren, P. AMCD: an accurate deep learning-based metallic corrosion detector for MAV-based real-time visual inspection. Journal of Ambient Intelligence and Humanized Computing, (2021), 1-12.

[13] Sudevan, V. Shukla, A. and Karki, H. Current and future research focus on inspection of vertical structures in oil and gas industry. In 18th International Conference on Control, Automation and Systems (ICCAS), 2018, pp. 144-149. IEEE

[14] Brandoli, B. de Geus, A. R. Souza, J. R. Spadon, G. Soares, A. Rodrigues Jr J. F., and Matwin, S. Aircraft fuselage corrosion detection using artificial intelligence. Sensors, (2021), 21(12), 4026.

[15] Rusakov, K. D. and Chekhov, A. V. Two-step Approach to Corrosion Detection of Metal Structures Using Convolutional Neural Networks When Inspecting Industrial Facilities. Proceedings of the Southwest State University, 2022, 25(3), 152–166. https://doi.org/10.21869/2223-1560-2021-25-3-152-166.

[16] Hoang, N. D. Image processing-based pitting corrosion detection using metaheuristic optimized multilevel image thresholding and machine-learning approaches. Mathematical Problems in Engineering, 2020, 1-19.

[17] Dhonde, K. Mirhassani, M. Tam, E. and Sawyer-Beaulieu, S. Design of a Real-Time Corrosion Detection and Quantification Protocol for Automobiles. Materials, (2022), 15(9), 3211.

[18] Yin, B. Josselyn, N. Considine, T. Kelley, J. Rinderspacher, B. Jensen, R.... and Site, A. N. R. E. (2021), Corrosion Image Data Set for Automating Scientific Assessment of Materials.

[19] M. Enikeev, I. Gubaydullin, and M. Maleeva, Analysis of corrosion process development on metals by means of computer vision. Engineering Journal, (2017), 21(4), 183-192.

[20] Yan, L. Diao, Y. Lang, Z. and Gao, K. Corrosion rate prediction and influencing factors evaluation of low-alloy steels in marine atmosphere using machine learning approach. Science and technology of Advanced Materials, (2020), 21(1), 359-370.

[21] Khayatazad, M. De Pue, L. and De Waele, W. Detection of corrosion on steel structures using automated image processing. Developments in the Built Environment, (2020), 3, 100022.

[22] Sumi, M. S. and Narayanan, A. Improving classification accuracy using combined filter+ wrapper feature selection technique. In 2019 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT), 2019, pp. 1-6). IEEE.

[23] Sun, X. Liu, Y. Xu, M. Chen, H. Han, J. and Wang, K. Feature selection using dynamic weights for classification. Knowledge-Based Systems, (2013), 37, 541-549.

[24] Karabulut, E. M. Özel, S. A. and Ibrikci, T. A comparative study on the effect of feature selection on classification accuracy. Procedia Technology, (2012), 1, 323-327.
[25] Duboue, P. The art of feature engineering: essentials for machine learning. Cambridge University Press. 2020.
[26] Baldi, P. Autoencoders, unsupervised learning, and deep architectures. In Proceedings of ICML workshop on unsupervised and transfer learning. JMLR Workshop and Conference Proceedings, 2012, June, pp. 37-49.
[27] Chen, Y. Zheng, B. Zhang, Z. Wang, Q. Shen, C. and Zhang, Q. Deep learning on mobile and embedded devices: State-of-the-art, challenges, and future directions. ACM Computing Surveys (CSUR), (2020), 53(4), 1-37.
[28] Deng, Y. Deep learning on mobile devices: a review. In Mobile Multimedia/Image Processing, Security, and Applications, SPIE, 2019, Vol. 10993, pp. 52-66.
[29] Prabhu, D. R. and Winfree, W. P. Neural Network Based Processing of Thermal NDE Data for Corrosion Detection. In Review of Progress in Quantitative Non-destructive Evaluation; Thompson, D.O., Chimenti, D.E., Eds.; Springer: Boston, MA, USA, 1993; Volume 12, pp. 775–782.
[30] Bellinger, N. C. and Komorowski, J. P. Corrosion pillowing stresses in fuselage lap joints. AIAA journal, (1997), 35(2), 317-320.
[31] Tan, X. M. Chen, Y. L.and Ping, J. I. N. Corrosion Fatigue Life Prediction of Aircraft Structure Based on Fuzzy Reliability Approach. Chin. J. Aeronaut, 2005,18, 346–351.
[32] Rusakov, K. D. and Chekhov, A. V. Two-step Approach to Corrosion Detection of Metal Structures Using Convolutional Neural Networks When Inspecting Industrial Facilities. Proceedings of the Southwest State University, 2022, 25(3), 152–166. https://doi.org/10.21869/2223-1560-2021-25-3-152-166.
[33] Lemos, R. Cabral, R. Ribeiro, D. Santos, R. Alves, V. and Dias, A. Automatic Detection of Corrosion in Large-Scale Industrial Buildings Based on Artificial Intelligence and Unmanned Aerial Vehicles. Applied Sciences, (2023), 13(3), 1386.
[34] Nash, W. Zheng, L. and Birbilis, N. Deep learning corrosion detection with confidence. npj Materials Degradation, (2022), 6(1), 26.
[35] Thompson, N. C. Greenewald, K. Lee, K. and Manso, G. F. The computational limits of deep learning. arXiv preprint arXiv:2007.05558, 2020.
[36] Goodfellow, I., Y. Bengio and Courville, A. Deep Learning. Cambridge, MA, USA: The MIT Press, 2016. Retrieved from https://www.deeplearningbook.org/
[37] Hinton, G. E. and Salakhutdinov, R. R. Reducing the dimensionality of data with neural networks. Science. 2006, 313, 5786, 504–507
[38] Wold, S. Esbensen, K. and Geladi, P. Principal component analysis. Chemometrics and Intelligent Laboratory Systems. 1987, 2, 37–52.
[39] Maas, A. L. Le, Q. V. O'Neil, T. M. Vinyals, O. Nguyen, P. and Ng, A. Y. Recurrent neural networks for noise reduction in robust ASR. INTERSPEECH, 2012, 22–25. Ma, X. Lin, Y. Nie, Z. and Ma, H. Structural damage identification based on unsupervised feature-extraction via Variational Auto-encoder. Measurement, (2020), 160, 107811.
[40] Ma, X. Lin, Y. Nie, Z. and Ma, H. Structural damage identification based on unsupervised feature-extraction via Variational Auto-encoder. Measurement, (2020), 160, 107811.
[41] Xing, C. Ma, L. and Yang, X. Stacked denoise autoencoder based feature extraction and classification for hyperspectral images. Journal of Sensors, 2016.
[42] Kingma, D. P. and Welling, M. An introduction to variational autoencoders. Foundations and Trends in Machine Learning, (2019), 12(4), 307-392.
[43] Higgins, I. Matthey, L. Pal, A. Burgess, C. Glorot, X. Botvinick, M. and Lerchner, A. beta-vae: Learning basic visual concepts with a constrained variational framework. In International conference on learning representations, 2017, April.
[44] Bosch, N. and Paquette, L. Unsupervised deep autoencoders for feature extraction with educational data. In Deep learning with educational data workshop at the 10th international conference on educational data mining, 2017, June. Kingma, D. P. and Welling, M. Auto-encoding variational bayes., 2013, arXiv:1312.6114.
[45] Kingma, D. P. and Welling, M. Auto-encoding variational bayes., 2013, arXiv:1312.6114.
[46] Cao, S. Li, J. Nelson, K. P. and Kon, M. A. Coupled VAE: Improved accuracy and robustness of a variational autoencoder. Entropy, (2022), 24(3), 423.
[47] Doersch, C. Tutorial on Variational Autoencoders. Pittsburgh, PA, USA: Carnegie Mellon University, 2016, Retrieved from https://arxiv.org/abs/1606.05908
[48] Zhang, M. Jiang, S. Cui, Z. Garnett, R. and Chen, Y. D-vae: A variational autoencoder for directed acyclic graphs. Advances in Neural Information Processing Systems, (2019), 32.
[49] Sønderby, C. K. Raiko, T. Maaløe, L. Sønderby, S. K. and Winther, O. 2016. Ladder Variational Autoencoders. Advances in Neural Information Processing Systems, 29, (2016), 3738–3746.
[50] Mumuni, A. and F. Mumuni, Data augmentation: A comprehensive survey of modern approaches. Array, (2022), 100258.
[51] Oza, P. Sharma, P. Patel, S. Adedoyin, F.and Bruno, A. Image augmentation techniques for mammogram analysis. Journal of Imaging, (2022), 8(5), 141.
[52] Frazier, P. I. Bayesian optimization. In Recent advances in optimization and modeling of contemporary problems. Informs, (2018), pp. 255-278.