Project Work

MTIR19

Final Report

Project Topic: Deep Learning Solution for Enhanced Additive Manufacturing Quality Control - Data Augmentation and Classification Model Development with DBN

Student Names:

1	Gokulnath V	112120018
2	Karan s	112120028

Year: 4th

Guide: Dr. M. Punniya Moorthy

Department: Department of Management studies

Internal Guide: Dr. S. Anand

Department: Metallurgical and Materials Engineering

Dr. M. Punniya Moorthy
(Project Guide)

Dr. S. Anand
(Internal Guide)

BONAFIDE CERTIFICATE

This is to certify that the project titled **Deep Learning Solution for Enhanced Additive Manufacturing Quality Control** is a bonafide record of the work done by

GOKULNATH V (112120018)

KARAN S (112120028)

in partial fulfilment of the requirements for the award of the degree of Bachelor of Technology in Metallurgical and Materials Engineering of the **NATIONAL INSTITUTE OF TECHNOLOGY, TIRUCHIRAPPALLI,** during the year 2023 –2024.

Dr. M. Punniyamoorthy
Professor (HAG)
Department of Management Studies
(Project Guide)

Dr. V. Karthick
Assistant Professor
Department of MME
(Internal Guide)

Project Viva-Voce held on __________________________

Internal Examiner

External Examiner

TABLE OF CONTENTS

ABSTRACT

Additive manufacturing (AM) processes have been receiving increasing research and industrial attention in recent years, as they are considered to revolutionize industrial production. However, one of the significant challenges faced by the AM process is the limited availability of high-quality datasets for training machine learning and deep learning models, particularly for classification tasks. In this project, we propose an approach to address this challenge by leveraging Deep Belief Networks (DBNs) for data augmentation and classification.

The primary objective of this project is to enhance the performance of classification models in additive manufacturing settings. To achieve this objective, we first train a DBN model using an existing dataset of AM images. The trained DBN model learns the underlying distribution of the image data and generates synthetic samples that closely resemble real AM images.

We compare the performance of classification models trained on varying sizes of datasets. Evaluation metrics such as the accuracy score and correlation coefficient are employed to assess the classification performance.

Augmenting the original dataset with DBN-generated samples should lead to significant improvements in classification accuracy and robustness. Furthermore, the augmented dataset enables the classification model to generalize better to unseen data and variations in the manufacturing process.

By mitigating the impact of dataset limitations, our method has the potential to facilitate the development of more accurate and reliable classification systems in manufacturing units, ultimately leading to improved quality control and efficiency.

INTRODUCTION

Additive Manufacturing (AM), commonly known as 3D printing, has emerged as a transformative technology in various industrial sectors, offering unprecedented flexibility and efficiency in production processes. The ability to fabricate complex geometries with high precision and customization has made AM an indispensable tool in aerospace, automotive, healthcare, and other fields. However, despite its potential, the widespread adoption of AM is hindered by challenges related to quality control, process optimization, and material characterization.

One of the critical factors influencing the quality and reliability of AM processes is the ability to accurately classify and identify defects in manufactured components. Traditional approaches to defect detection often rely on manual inspection or rule-based systems, which are time-consuming, labor-intensive, and prone to human error. In contrast, machine learning techniques offer a promising avenue for automating defect detection and quality assurance in AM.

The success of machine learning models in AM applications crucially depends on the availability of high-quality datasets for training and validation. However, acquiring large and diverse datasets for AM is a daunting task due to factors such as cost, time, and proprietary restrictions. Consequently, researchers and practitioners often face the challenge of limited data, which can adversely affect the performance and generalization capability of machine learning models.

To address the data scarcity problem in AM, this project proposes a novel approach that leverages Deep Belief Networks (DBNs) for data augmentation and classification. DBNs, a type of generative neural network architecture, have demonstrated remarkable capabilities for learning complex data distributions and generating realistic samples. By harnessing the power of DBNs, we aim to augment the existing dataset of AM images and improve the performance of classification models for defect detection and quality control.

The architecture of the DBN model utilized in this research was developed using the PyTorch framework. The implementation incorporates various techniques, including Contrastive Divergence (CD), Gibbs sampling, and cross-entropy optimization, to train the DBN model effectively. CD and Gibbs sampling are used to approximate the intractable posterior distribution of the model's parameters, while cross-entropy optimization is employed to minimize the reconstruction error and fine-tune the model parameters.

In this study, we utilize an image dataset sourced from https://data.mendeley.com/datasets/zyz6cznm5h/3, which comprises a diverse collection of Artifact images AM through Fused Deposition Modelling. The dataset serves as the foundation for training the DBN model and evaluating the performance of the proposed approach.

The primary objective of this research is to develop DBN model from scratch, followed by an investigation of the feasibility and effectiveness of using DBNs for data augmentation in AM datasets and an evaluation of the impact of dataset size on the classification performance of DBN models. We aim to demonstrate the potential of our approach in overcoming the challenges of limited data and improving the reliability of defect detection systems in AM.

In the upcoming sections, we delve into the methodology employed in this research, detailing the architecture and training procedure of the DBN model, as well as the experimental setup and evaluation metrics used to assess classification performance. Finally, we present the results of our model and discuss their implications.

LITERATURE REVIEW

In recent years, the application of deep learning techniques has witnessed significant advancements, particularly in addressing challenges related to limited dataset availability and improving classification performance in manufacturing processes. This section presents a review of relevant literature, focusing on studies that explore the use of Deep Belief Networks (DBNs) and related techniques for data augmentation and classification tasks in additive manufacturing.

Supervised Deep Learning for Manufacturing Quality Prediction

This paper introduces a methodology for quality prediction in industrial processes using Supervised Deep Belief Networks (SDBNs). Their research demonstrated the efficacy of SDBNs in capturing complex relationships between process variables and quality outcomes, showcasing the potential of deep learning models for predictive maintenance and quality assurance in manufacturing settings. The insights gained from their study lay the groundwork for leveraging DBNs in data augmentation and classification tasks to enhance the accuracy and reliability of quality prediction models in additive manufacturing.

Real-time Quality Monitoring and Diagnosis with Deep Belief Networks

This paper proposes a real-time quality monitoring and diagnosis framework based on Deep Belief Networks (DBNs) for manufacturing process profiles. By harnessing the hierarchical representation learning capabilities of DBNs, their approach enables the timely detection and diagnosis of quality deviations in additive manufacturing processes. The integration of DBNs into the monitoring and diagnosis framework offers a promising avenue for improving process control and quality assurance in additive manufacturing operations, thereby enhancing the overall efficiency and reliability of the manufacturing process.

Quality Analysis in Additive Manufacturing using Deep Learning

This paper investigated the application of deep learning techniques, including Deep Neural Networks (DNNs), for quality analysis in metal additive manufacturing processes. Their research focused on leveraging DNNs to analyze complex microstructural features and defect patterns in additively manufactured components. By utilizing deep learning models trained on large-scale image datasets, it demonstrated the potential of automated defect detection and quality assessment in additive manufacturing, highlighting the role of deep learning in augmenting traditional quality control practices.

Deep Learning for Smart Manufacturing: Methods and Applications

This paper provides a comprehensive review of deep learning methods and applications in smart manufacturing systems. Their study underscored the diverse range of deep learning techniques employed across various manufacturing domains, including process monitoring, quality control, and predictive maintenance. The integration of deep learning into smart manufacturing systems offers opportunities for enhancing production efficiency, product quality, and resource utilization, aligning with the overarching goal of leveraging advanced technologies to optimize additive manufacturing processes.

METHODOLOGY

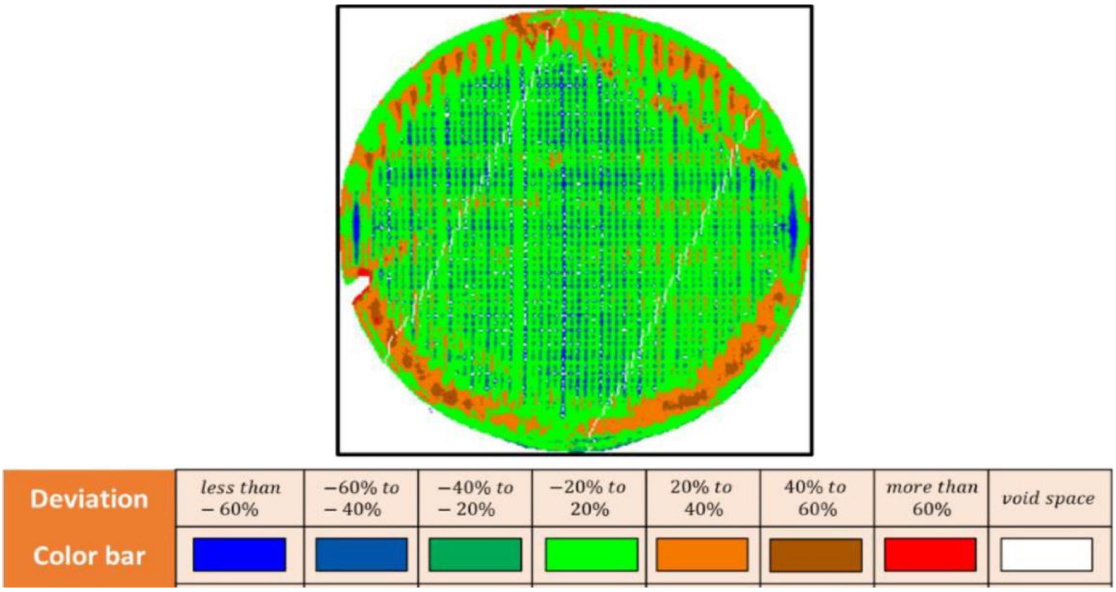
1) Dataset Description and Pre-processing

a) Dataset

The dataset used in this project consists of 43400 images from the top surface of the artifacts which was Additively manufactured using Fused Deposition Modeling (FDM). Each image is pre-analyzed and threshold to generate color map data to represent the surface quality better. The images was labeled by professionals into four categories: (a) Over Printing Situation, (b) Normally Printed Situation, (c) Under Printing Situation, (d) Empty.

A high-speed 2D Laser Profiler KEYENCE LJ-V7000 series is equipped above the FDM machine.

The color mapping rule used in the dataset is shown below.



b) Data Pre-processing

The pre-processing steps include converting the images to grayscale, resizing them to a uniform size, and converting them to tensor format suitable for training with PyTorch.

Grayscale Conversion:

The ‘transforms.Grayscale()’ function is used to convert the RGB images to grayscale. Grayscale images contain only intensity values ranging from 0 to 255, which simplifies the computational complexity of the model while preserving essential features for classification.

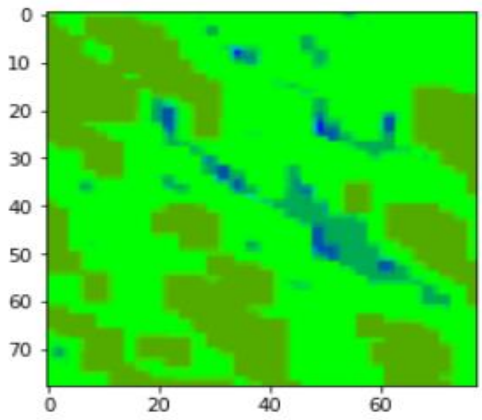
Image Resizing:

The ‘transforms.Resize((78, 78))’ function resizes the images to a uniform size of 78x78 pixels. Standardizing the image size ensures consistency in the input dimensions across the dataset, facilitating efficient training of the DBN model.

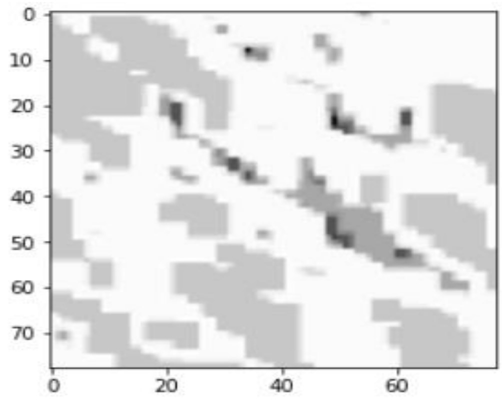
Tensor Conversion:

The ‘transforms.ToTensor()’ function converts the preprocessed images into tensor format, which is compatible with PyTorch's data processing pipeline.

Original Image



Pre-Processed Image



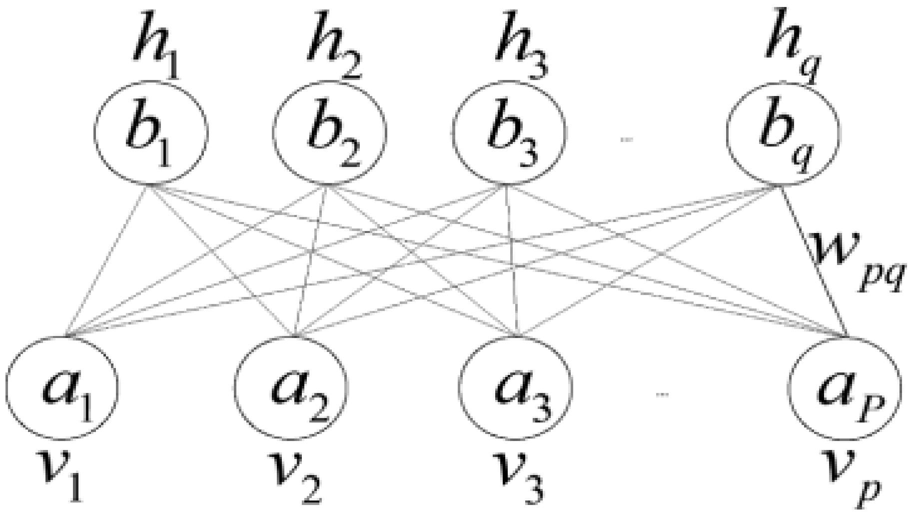
Tensor Format of above image:

tensor ([0.5725, 0.5725, 0.5882, 0.5882, 0.5882, ..., 0.5882, 0.5882, 0.5882])

By applying these pre-processing steps, we ensure that the input data is appropriately formatted and standardized for training the DBN model.

2) **DBM Model Architecture**

Our Deep Belief Network (DBN) model architecture consists of two layers of Restricted Boltzmann Machines (RBMs) stacked together, followed by a fully connected layer for classification. The RBMs serve as building blocks for learning hierarchical representations of the input data, while the fully connected layer integrates the learned features for classification.



Architecture of RBM (v is the input visible vector and h is the hidden feature vector)

Energy Function:

The energy function, denoted by $E(v,h)$, is a fundamental concept in the RBM model. It represents the interaction between visible units v and hidden units h in the RBM. The energy function is defined as a function of the states of both visible and hidden units, along with the parameters of the model (weights and biases).

Mathematically, the energy function is expressed as:

$$E(v,h) = -\sum_{i \in n_v, j \in n_h} w_{ij} v_i h_j - \sum_{i \in n_v} a_i v_i - \sum_{j \in n_h} b_j h_j$$

n_v and n_h represent the dimension of the corresponding visible vector v and hidden vector h, respectively; $v_i(i = 1, 2, \ldots , nv)$ and $h_j(j = 1, 2, \ldots , nh)$ represent unit i of the visible layer and unit j of the hidden layer, respectively;

a_i and b_j are the corresponding bias terms of the visible unit v_i and the hidden unit h_j , respectively; w_{ij} is the connection weight between v_i and h_j.

Joint Distribution:

The joint distribution p(v,h) represents the probability of observing a specific configuration of visible and hidden units in the RBM. It is derived from the energy function using the Boltzmann distribution. The joint distribution is defined as:

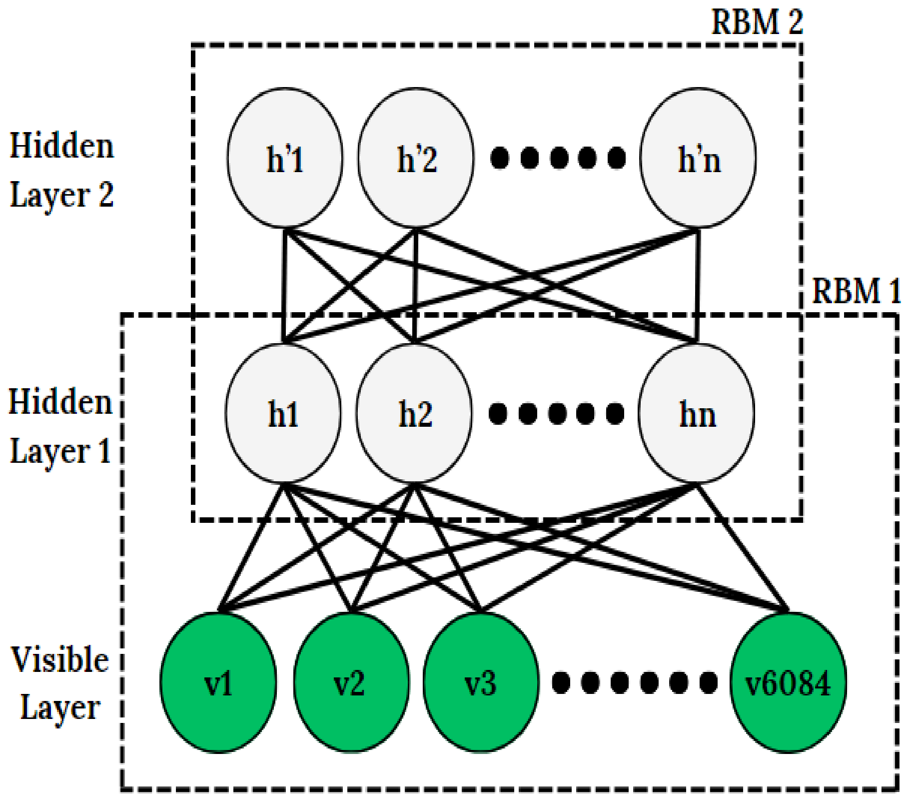
$$p(v, h) = \frac{e^{-E(v,h)}}{Z}$$

Where Z is the normalization constant, also known as the partition function, given by the sum of the probabilities of all possible configurations of visible and hidden units:

$$Z = \sum_{v,h} e^{-E(v,h)}$$

The joint distribution encapsulates the overall probability distribution of the RBM's state space. It governs the likelihood of different configurations of visible and hidden units occurring in the RBM.

Our Model Architecture:



3) Training

a) Pre-training with RBM's

Pre-training is a crucial step in training Deep Belief Networks (DBNs), where the parameters of each layer are initialized using unsupervised learning techniques. Pre-training allows the DBN to learn useful hierarchical representations of the input data, which can facilitate subsequent fine-tuning for specific tasks such as classification.

The pre-training process involves training each RBM layer-by-layer in an unsupervised manner. The RBM learns to capture complex dependencies in the data by minimizing the reconstruction error between the input data and the reconstructed data generated by the model.

Contrastive Divergence (CD)

Contrastive Divergence (CD) is a popular algorithm used for pre-training RBMs. It approximates the gradient of the log-likelihood function with respect to the model parameters using a Markov chain Monte Carlo (MCMC) sampling procedure. CD consists of the following steps:

Initialization: Initialize the visible units with the input data.

Gibbs Sampling: Perform alternating Gibbs sampling to generate samples of hidden units and visible units.

Hidden Units given Visible Units:

$$P(h_j = 1|v) = \sigma\left(b_j + \sum_{i=1}^{N_v} W_{ij} v_i\right)$$

Visible Units given Hidden Units:

$$P(v_i = 1|h) = \sigma\left(a_i + \sum_{j=1}^{N_h} W_{ij} h_j\right)$$

Positive Phase: Compute the positive gradient using the product of the input data and the sampled hidden units.

$$v_0^T h_0$$

Negative Phase: Compute the negative gradient using the product of the reconstructed visible units and the sampled hidden units.

$$v_1^T h_1$$

Parameter Update:

- Weight Update:

$$\Delta W = \alpha(v_0^T h_0 - v_1^T h_1)$$

- Visible Bias Update:

$$\Delta b = \alpha(v_0 - v_1)$$

- Hidden Bias Update:

$$\Delta c = \alpha(h_0 - h_1)$$

Where: α is the learning rate.

In our project, we pre-train each layer of the DBN using RBMs with Contrastive Divergence. We initialize the parameters of each RBM using random Gaussian initialization. The pre-training process consists of multiple iterations of Contrastive Divergence, with a fixed learning rate and batch size.

b) Fine-Tuning

Fine-tuning is a crucial step in training Deep Belief Networks (DBNs) where the pre-trained parameters are further optimized using labelled data and supervised learning techniques. Fine-tuning enables the DBN to adapt its learned representations to the specific task at hand, such as classification, by adjusting the parameters to minimize the classification error.

Cross-Entropy Loss Function

In fine-tuning the DBN for classification tasks, the cross-entropy loss function is commonly used to measure the discrepancy between the predicted probabilities and the true labels. The cross-entropy loss is defined as:

$$L(y, \hat{y}) = -\frac{1}{N} \sum_{i=1}^{N} \sum_{c=1}^{C} y_{i,c} \log(\hat{y}_{i,c})$$

Where:

N is the number of samples,

C is the number of classes,

$y_{i,c}$ is the true label (1 if sample i belongs to class c, 0 otherwise), and

$y^{\wedge}{}_{i,c}$ is the predicted probability that sample i belongs to class C.

Minimizing the cross-entropy loss encourages the model to assign high probabilities to the correct class labels and low probabilities to incorrect class labels.

Backpropagation

Fine-tuning the DBN involves using backpropagation with gradient descent to update the parameters of the network. During fine-tuning, the gradients of the cross-entropy loss with respect to the model parameters are computed using backpropagation, and the parameters are updated in the direction that minimizes the loss.

In our experiments, we fine-tune the pre-trained DBN using labelled data from the training set. The fine-tuning process consists of multiple epochs, where each epoch involves iterating over mini-batches of labelled data, computing the gradients of the cross-entropy loss with respect to the parameters, and updating the parameters.

c) Hyper parameters Tuning

Hyper parameters tuning is a critical aspect of training Deep Belief Networks (DBNs) to achieve optimal performance. Hyper parameters are parameters that are not learned directly from the data but rather control the learning process, such as the learning rate, batch size, and number of epochs. Tuning these hyper parameters effectively can significantly impact the performance and convergence of the DBN model.

<u>Learning Rate</u>

The learning rate determines the size of the steps taken during optimization and affects the convergence speed of the model. A high learning rate may lead to unstable training, while a low learning rate may result in slow convergence. Tuning the learning rate involves selecting an appropriate value that balances convergence speed and stability.

<u>Batch Size</u>

The batch size determines the number of samples processed in each iteration of training. A larger batch size can lead to faster convergence but may require more memory and computational resources. Conversely, a smaller batch size may result in slower convergence but can provide better generalization. Tuning the batch size involves finding the optimal balance between convergence speed and generalization performance.

<u>Number of Epochs</u>

The number of epochs defines the number of times the entire training dataset is passed through the model during training. Too few epochs may result in under fitting, while too many epochs may lead to overfitting.

Tuning the number of epochs involves selecting an appropriate value that allows the model to converge without overfitting to the training data.

We evaluate the performance of the DBN model for each hyper parameter configuration using test dataset. The selected hyper parameters are then used to train the final DBN model on the entire training dataset.

4) Data Augmentation

Data augmentation is a technique used to increase the size and diversity of the training dataset by applying various transformations to the original data samples. In the context of Deep Belief Networks (DBNs), data augmentation can help improve the generalization and robustness of the model by exposing it to a wider range of variations in the input data.

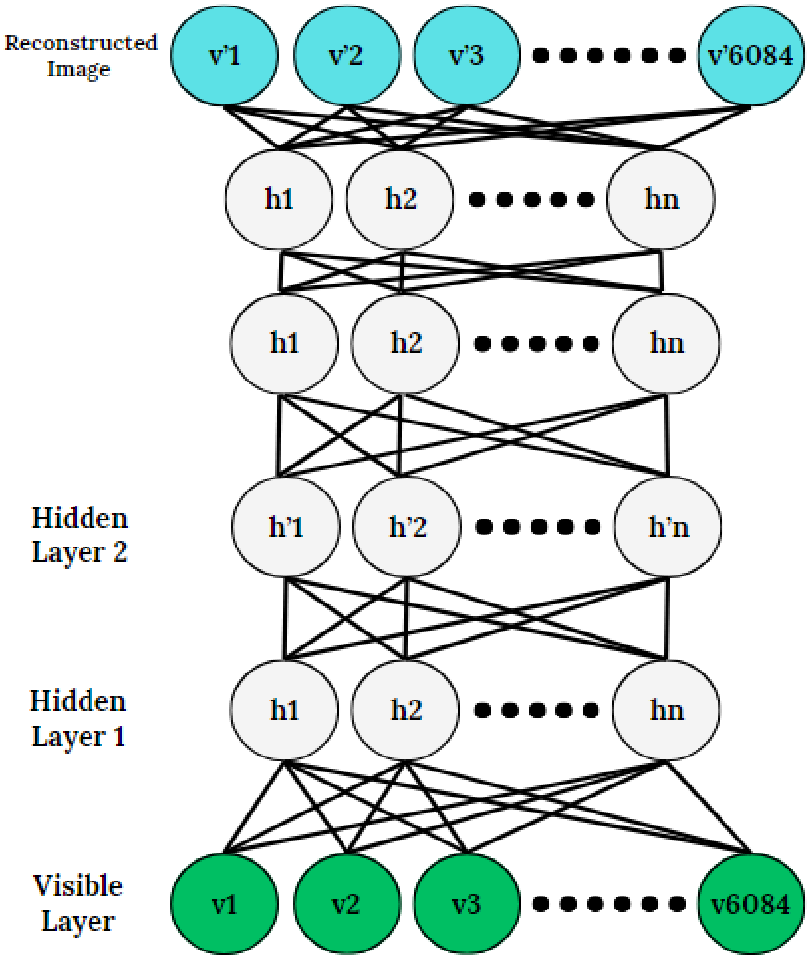
Image Generation

One approach to data augmentation in DBNs is to generate synthetic images using the trained DBN model. After pretraining the DBN on the original dataset, the model can generate new samples by sampling from the learned distribution of hidden unit activations. These generated images can then be added to the training dataset to increase its size and diversity.

Sampling Technique

Sampling from the learned distribution of hidden unit activations in the DBN involves performing Gibbs sampling technique. Gibbs sampling alternates between sampling the states of visible and hidden units based on the probability distributions defined by the RBM's energy function. By repeatedly sampling from the learned distribution, the DBN can generate realistic and diverse samples that capture the underlying structure of the input data.

DBN Model Architecture for Image Generation:



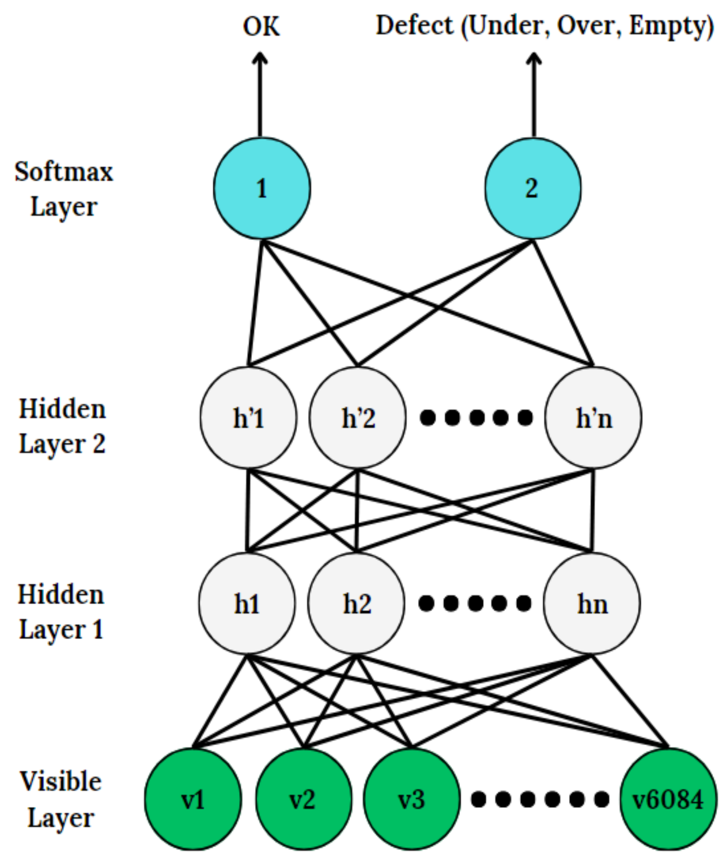
5) Classification Model

The classification model in our thesis builds upon the pretrained Deep Belief Network (DBN) by adding a fully connected layer for classification.

Model Architecture

The classification model extends the pretrained DBN by appending a fully connected layer at the top, followed by a softmax activation function. The fully connected layer maps the learned features from the DBN to the output classes, enabling the model to perform classification.

DBN Model Architecture for Classification:



The architecture of the classification model is as follows:

1. DBN Layers: The pretrained DBN layers, which capture hierarchical representations of the input data.

2. Fully Connected Layer: A full connected layer that takes the output of the DBN as input and maps it to the output classes.
3. Softmax Activation: A softmax activation function applied to the output of the fully connected layer to compute the probability distribution over the output classes.

Training

The classification model is trained using labeled data from the training set. We use the cross-entropy loss function to measure the discrepancy between the predicted probabilities and the true labels. The model parameters, including the weights and biases of the fully connected layer, are optimized using stochastic gradient descent (SGD) with momentum.

During training, we iterate over mini-batches of labeled data and compute the gradients of the cross-entropy loss with respect to the model parameters using backpropagation. The parameters are updated in the direction that minimizes the loss, gradually improving the model's performance on the classification task.

Classification

Once the image has passed through all the layers of the DBN, the final layer typically consists of a fully connected layer followed by a softmax activation function. The fully connected layer integrates the learned features from the preceding layers and maps them to the output classes. The softmax activation function then computes the probability distribution over the output classes based on the integrated features.

Prediction

The class label corresponding to the highest probability in the softmax output is selected as the predicted class for the input image. The predicted class label indicates the category to which the input image is classified by the DBN model.

Performance Metrics

We evaluate the performance of the classification model using performance metrics, which includes:

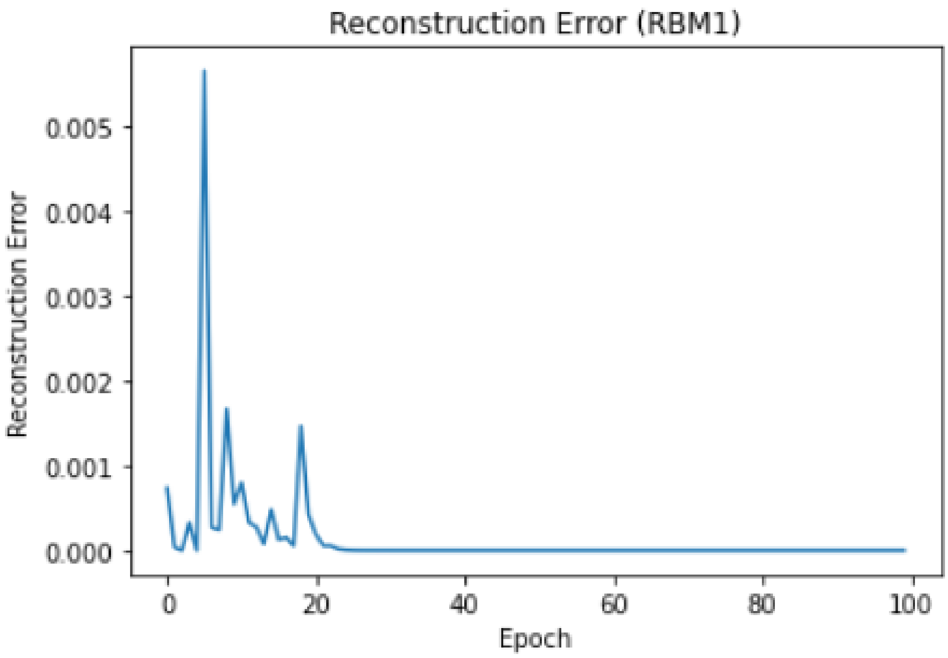
- **Accuracy**: The proportion of correctly classified samples out of the total number of samples.

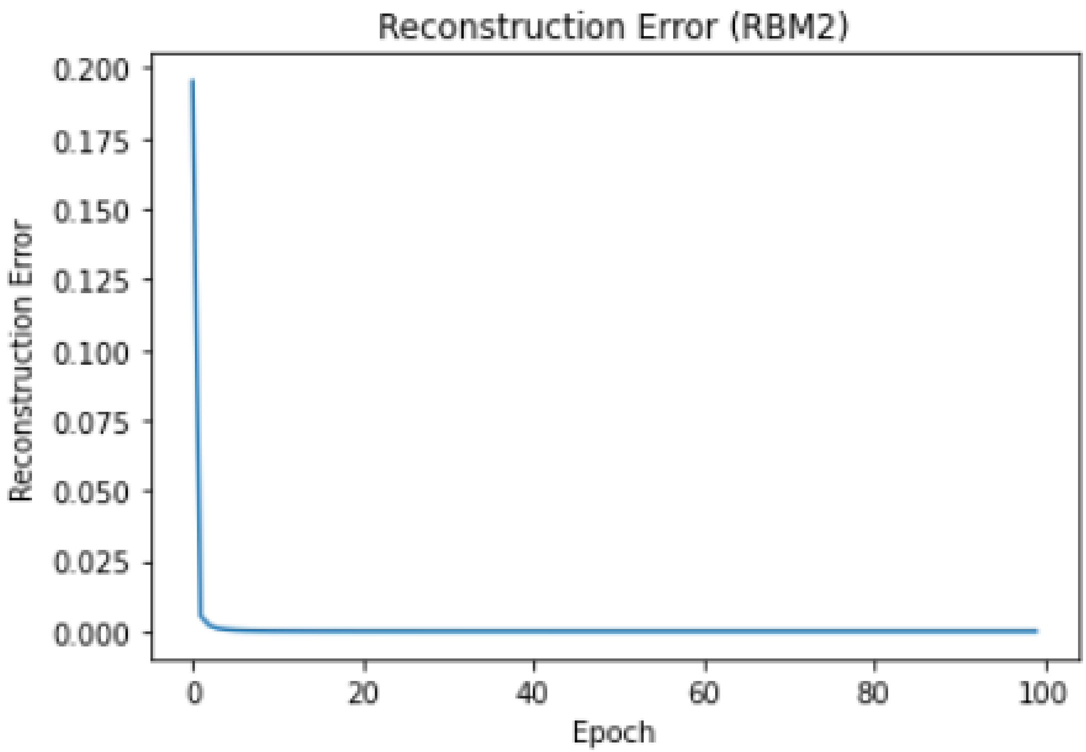
These performance metrics provide insights into the classification performance of the model, including its overall accuracy and its ability to correctly classify samples belonging to each class.

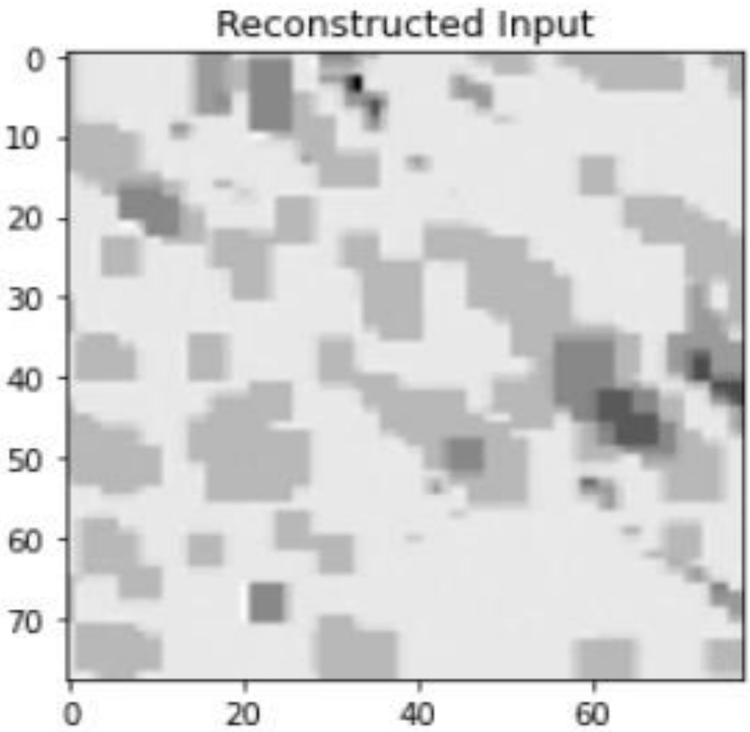
RESULTS & DISCUSSION

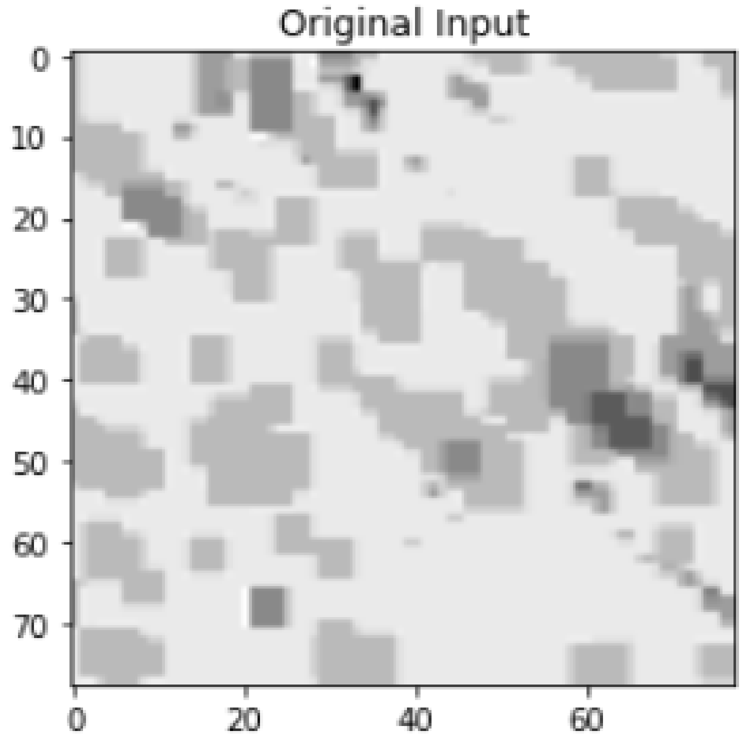
1) Data Generation

a) DBN Trained with one Image









Mean Squared Error (MSE): 1.1108273838544847e-06

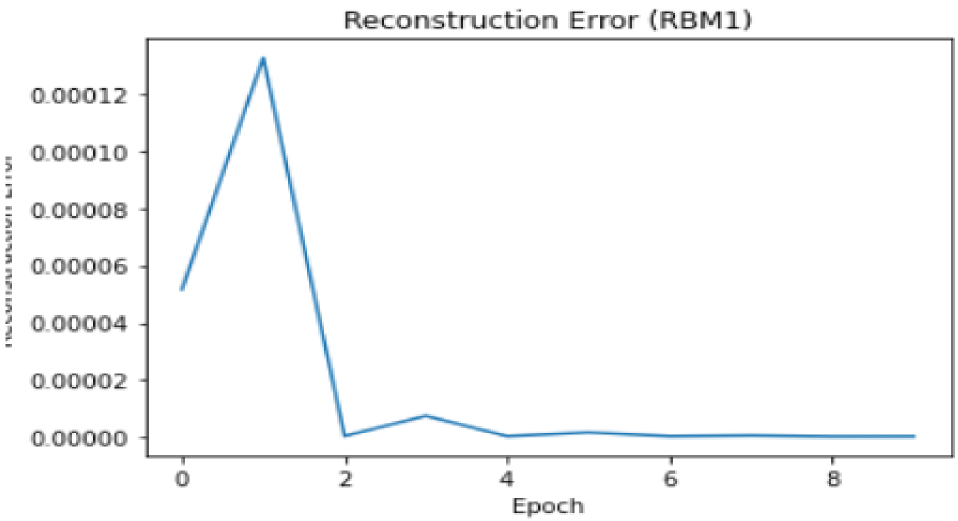
Structural Similarity Index (SSIM): 0.99890877783160591

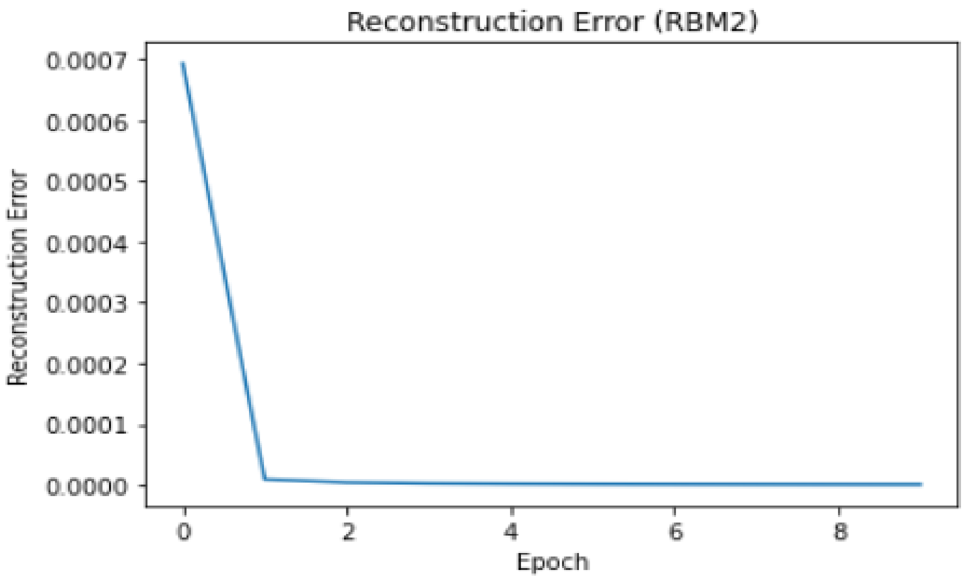
Correlation coefficient: 0.9999276524059919

b) DBN Trained with Dataset

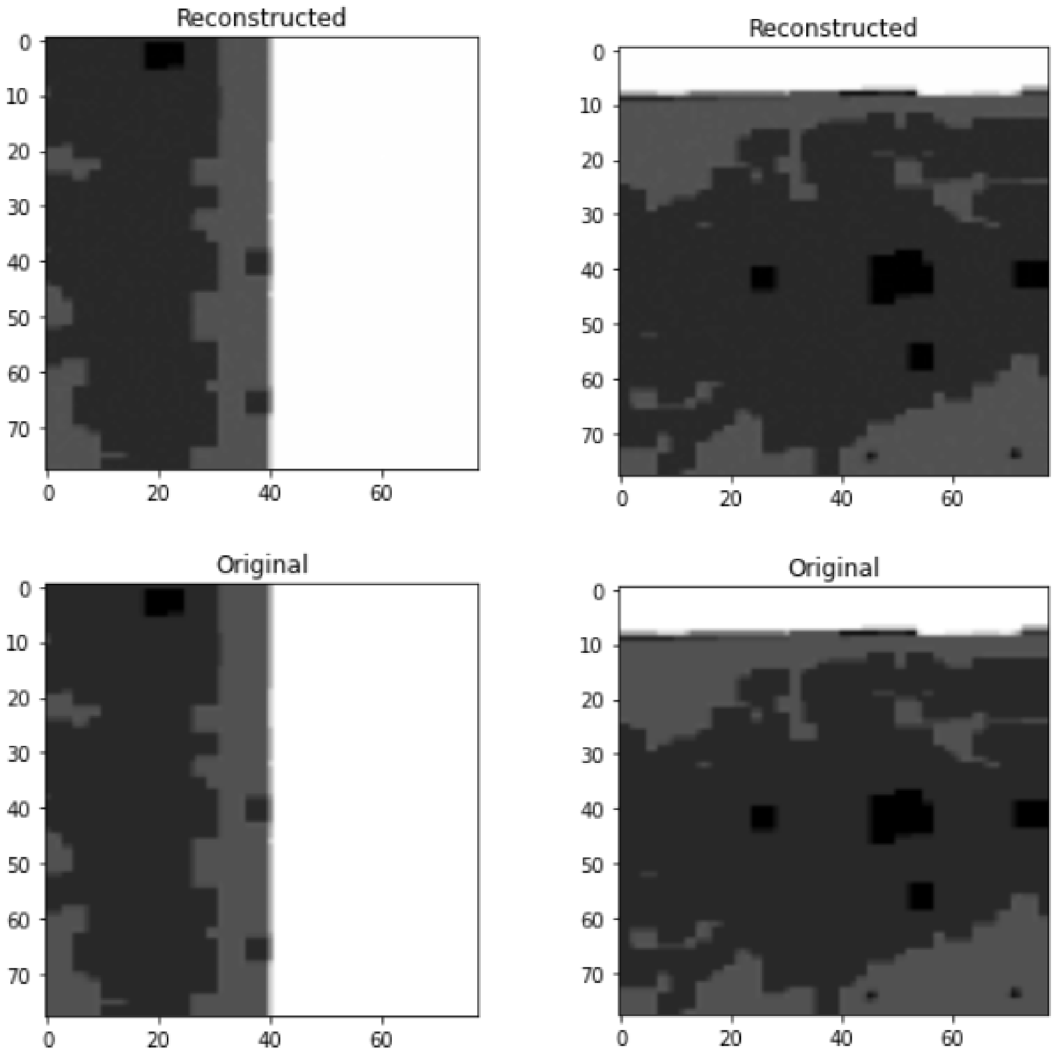
The first step in Image Augmentation is to train the DBN model with limited dataset, followed by regenerate the images using trained model.

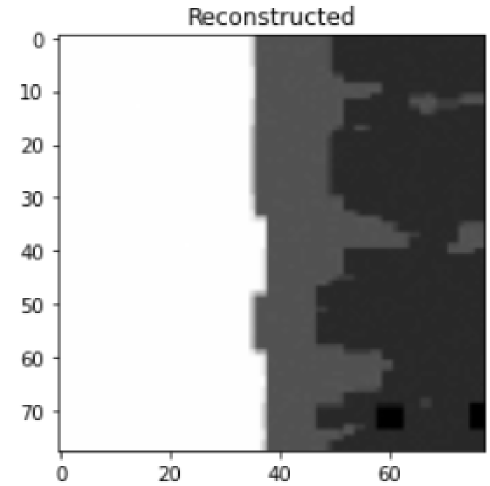
The RBMs were trained using a contrastive divergence algorithm, where the weights and biases were updated to minimize the reconstruction error. The training process involved iteratively training each RBM layer while updating the weights and biases based on the reconstruction error.

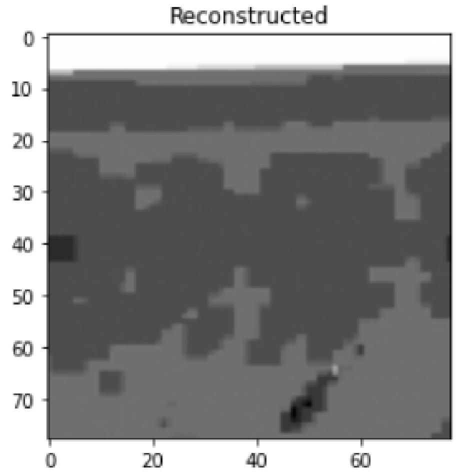


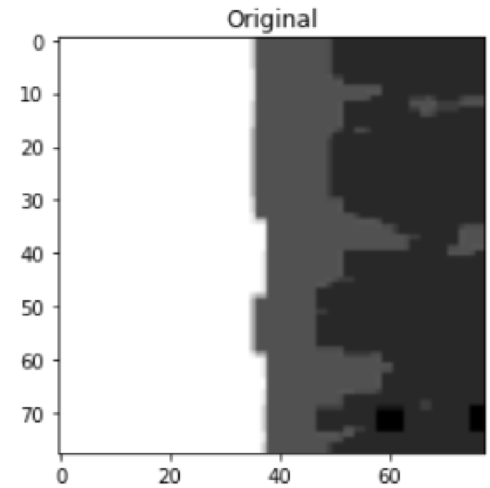


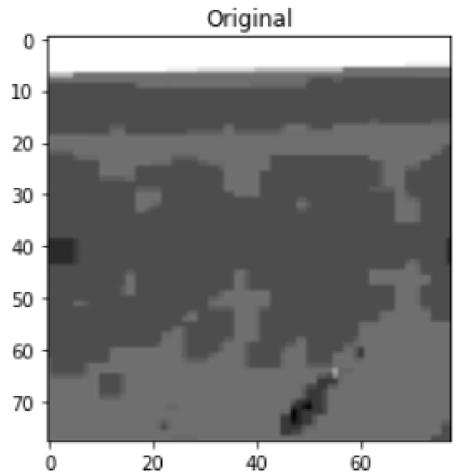
Examples:

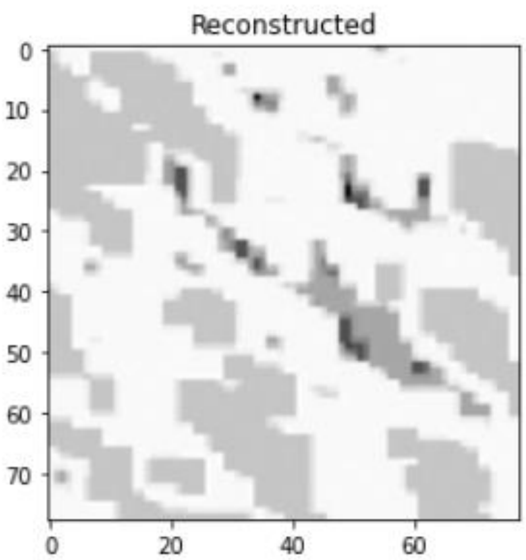

Reconstructed
Reconstructed
Original
Original
0
10
20
30
40
50
60
70
0
20
40
60

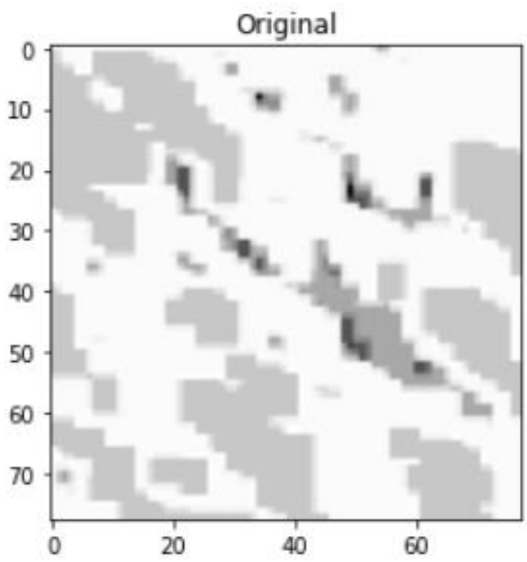

Reconstructed
0 10 20 30 40 50 60 70
0 20 40 60


Reconstructed
0 10 20 30 40 50 60 70
0 20 40 60

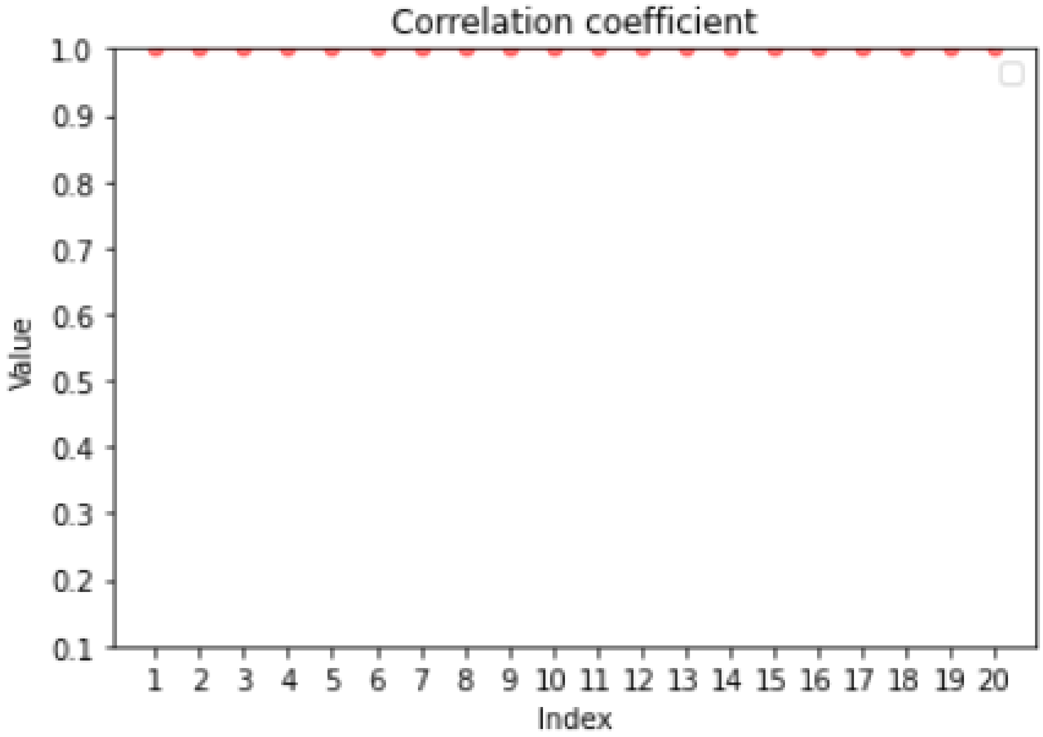

Original
0 10 20 30 40 50 60 70
0 20 40 60


Original
0 10 20 30 40 50 60 70
0 20 40 60


Reconstructed
0 10 20 30 40 50 60 70
0 20 40 60


Original
0 10 20 30 40 50 60 70
0 20 40 60

To assess the quality of the generated images over entire dataset, we calculated correlation coefficient for each images with its respective regenerated images over the entire dataset. Overall, the regenerated images closely resemble the original images in terms of visual appearance and structural integrity.



The successful generation of images using the trained DBN highlights the model's ability to learn meaningful representations from the dataset. The regenerated images not only resemble the original images but also capture the dataset's intrinsic characteristics and variations. These results demonstrate the potential of DBNs as generative models for image data and open up avenues for applications in image synthesis and data augmentation.

c) Image generation using Trained DBN

Image generation using a trained Deep Belief Network (DBN) involves utilizing the learned representations and generative capabilities of the model to synthesize new images. After pretraining the DBN on the original dataset, the model can generate novel samples by sampling from the learned distribution of hidden unit activations.

Image generation using a trained DBN offers several benefits for data augmentation:

Increased Dataset Size: Generating synthetic images effectively increases the size of the training dataset, providing the model with more data to learn from and improving its generalization performance.

Diversity: The generated images capture the diversity of the input data, including variations and patterns that may not be present in the original dataset, enhancing the model's ability to generalize to unseen data.

Robustness: Exposure to a wider range of variations and perturbations through image generation improves the model's robustness to noise and variations in the real-world data, leading to better performance in practical applications.

2) Image Classification

The Deep Belief Network (DBN) model which has been trained and fine-tuned can be used for image classification.

Dataset 1:

Train dataset size: 80 Images

Test dataset size: 20 Images

Accuracy of the network on training 80 images: 70 %

Dataset 2:

Train dataset size: 160 Images

Test dataset size: 40 Images

Accuracy of the network on the 80 training images: 71 %

Dataset 3:

Train dataset size: 800 Images

Test dataset size: 200 Images

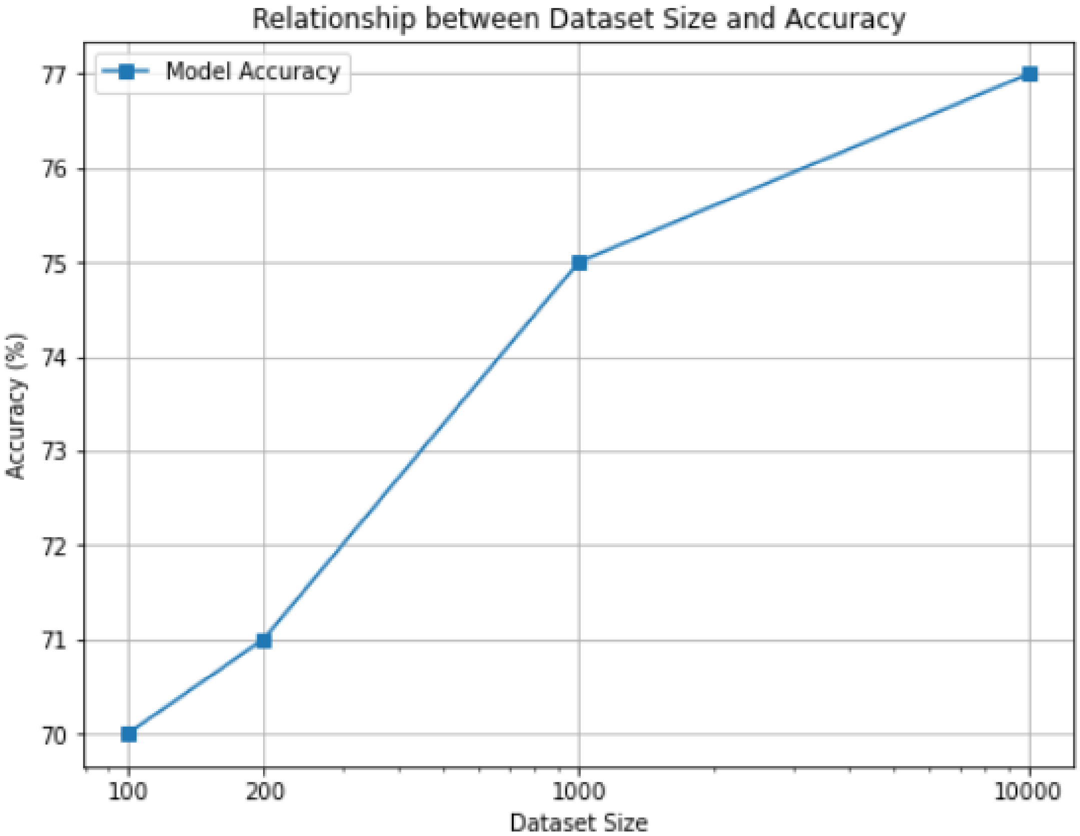
Accuracy of the network on training 80 images: 75 %

Dataset 4:

Train dataset size: 8000 Images

Test dataset size: 2000 Images

Accuracy of the network on training 80 images: 77 %



The above Graphs shows the model's performance on various benchmark datasets.

CONCLUSION

In this project, we have explored the application of Deep Belief Networks (DBNs) in addressing the challenge of limited datasets in additive manufacturing. The objective was to develop a DBN model for Image Regeneration, Generation and Image classification, followed by understanding how accuracy changes depending on the size of training dataset.

By leveraging the capabilities of DBNs to generate realistic synthetic samples, we were able to significantly expand the diversity and size of the original dataset. This augmentation, in turn, would enhanced the robustness and generalization capabilities of the classification model.

Our results indicate that the model trained with large dataset size leads to improved performance in classifying input images. This finding underscores the potential of data augmentation techniques in mitigating the challenges posed by limited datasets, particularly in industrial settings where data collection may be constrained.

Moreover, the implications of our research extend beyond the realm of image classification in additive manufacturing. The methodologies and insights presented here offer a blueprint for leveraging AI techniques to enhance process efficiency, quality control, and defect detection across various manufacturing domains.

By harnessing the power of Deep Belief Networks and data augmentation techniques, we pave the way for more resilient, adaptable, and efficient manufacturing systems in the years to come.

A potential future work may focus on the development of model with improved prediction performance and synthetic image generation capability of DBN.

REFERENCES

1. LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), 436-444.
2. Hinton, G. E., Osindero, S., & Teh, Y. W. (2006). A fast learning algorithm for deep belief nets. Neural computation, 18(7), 1527-1554.
3. Goodfellow, I., Bengio, Y., Courville, A., & Bengio, Y. (2016). Deep learning (Vol. 1). MIT press Cambridge.
4. Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. Journal of Big Data, 6(1), 1-48.
5. Giselsson, T. M., Andersson, H., & Tibert, G. (2019). Machine learning for additive manufacturing - where are we now and what are the opportunities?. Additive Manufacturing, 27, 489-510.
6. Tran, T., Phung, S. L., & Venkatesh, S. (2017). Deep learning for industrial applications. Springer.
7. Cai, S., & Wang, X. (2019). A review of data-driven approaches for quality control in additive manufacturing. Journal of Manufacturing Systems, 50, 76-95.
8. Wang, B., Rong, Y., Li, X., Wang, X., & Chu, F. (2020). A novel online monitoring method for additive manufacturing using deep learning. Additive Manufacturing, 34, 101317.