A Project Report

On

ANOMALY DETECTION IN MACHINES

BY

B V SOMA ADITHYA – SE21UARI192

ROHITH SIDDI – SE21UARI120

KOUSHIK D – SE21UARI162

JASMITHA BOLLA – SE21UARI028

Under the supervision of

Dr. DIPTI MISHRA

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR

DEGREE OF BACHELOR OF TECHNOLOGY

PR 3201: PROJECT TYPE COURSE

ECOLE CENTRALE SCHOOL OF ENGINEERING

MAHINDRA UNIVERSITY

HYDERABAD

(JUNE 2024)

ACKNOWLEDGMENTS

We would like to express my deepest gratitude to all those who contributed to the completion of this project. First and foremost, We are profoundly grateful to Dr. Dipti Mishra for her invaluable guidance, insightful feedback, and unwavering support throughout the course of this project.

We extend our heartfelt thanks to our colleagues and peers who offered constructive criticism, shared resources, and provided a motivating and collaborative environment.

Thank you, Professor Dipti Mishra, for your patience, wisdom, and for being a beacon of knowledge. We deeply appreciate the opportunity to learn from you.

Ecole Centrale School of Engineering

Mahindra University

Hyderabad

Certificate

This is to certify that the project report entitled "**Anomaly Detection in Machines**" submitted by Soma Adithya(SE21UARI192), Jasmitha (SE21UARI028), Koushik (SE21UARI162), Rohit (SE21UARI120) in partial fulfillment of the requirements of the course PR 3201, Project Course, embodies the work done by him/her under my supervision and guidance.

(Dr. Dipti Mishra & Signature)

Ecole Centrale School of Engineering, Hyderabad.

Date: 10^{th} JUNE 2024

ABSTRACT

In our Project, we developed an unsupervised anomaly detection model utilizing a Convolutional Autoencoder (CAE) to distinguish between normal and anomalous images in machines. The CAE was trained exclusively on normal images from the MVTec AD dataset, learning to compress and subsequently reconstruct the input images. We employed a novel combined loss function that integrates Mean Squared Error (MSE) and Structural Similarity Index (SSIM), enhancing the model's ability to accurately reconstruct normal images while amplifying reconstruction errors for anomalies. Our approach demonstrates the efficacy of CAEs in unsupervised anomaly detection, providing a reliable method for detecting defects in industrial applications.

CONTENTS

Title page……………………………………………………… …….1

Acknowledgements……………………………………………….2

Certificate…………………………………………………….....3

Abstract……………………………………………………....4

1.Introduction……...…………………….……….……… …....6

2.Data Set and Preprocessing…………………….…………..7

3.Model and Implementation

- 3.1 Model Architecture……………………………………….8
- 3.2 Implementation…………………………………………..9-10

4.Evaluation and Results……………………....…………11

Conclusion…………………………………………………..12

References………………………………………………….. 13

Figures used in the Report.

Fig 1	Encoder
Fig 2	Decoder
Fig 3	Histogram of training Reconstruction Errors
Fig 4	Train and Validation Loss

1. Introduction

In the contemporary industrial landscape, the detection of anomalies in machinery is paramount to maintaining operational efficiency, ensuring safety, and minimizing downtime. Anomalies in machines can arise from a multitude of factors, including wear and tear, manufacturing defects, improper maintenance, or unexpected environmental conditions. Detecting these anomalies early can prevent minor issues from escalating into major failures, which can lead to costly repairs, production halts, and potential safety hazards.

Traditional methods of anomaly detection in industrial settings often rely on manual inspections, predefined rules, or sensor-based monitoring systems. While these methods have their merits, they are frequently limited by their dependency on prior knowledge of possible anomalies, the need for extensive domain expertise, and the challenge of handling complex, high-dimensional data generated by modern machines.

This project leverages the capabilities of CAEs to develop an effective anomaly detection system for industrial machinery. By training on normal operational data, the CAE learns to reconstruct this data with minimal error. When the model encounters anomalous data, it produces higher reconstruction errors, signaling potential issues in the machinery. The MVTec Anomaly Detection (MVTec AD) dataset, which consists of real industrial images, is used to evaluate the model’s performance.

2. Data set and Preprocessing

For this project, we utilized the MVTec Anomaly Detection (MVTec AD) dataset, which is widely recognized for its comprehensive representation of industrial anomaly detection scenarios. The dataset includes high-resolution images across various categories such as "bottle," "hazelnut," "leather," and many others, each with different types of anomalies like scratches, dents, and foreign objects. To streamline the data for our analysis, we consolidated the original images into two primary subfolders. These subfolders represent the categories of normal and anomalous images. This organizational structure facilitated easier management and processing of the data, enabling a more efficient workflow for training and testing our anomaly detection models.

The image size of the MVTec AD are 900x900 which is very high. Due to computational constraints and the need for efficient processing, we resized these images to smaller dimensions. Specifically, we experimented with resizing the images to 64x64 and 128x128 pixels. After comparing different sizes, we selected 64x64 pixels as the optimal size for our model due to the balance it provided between computational efficiency and retention of critical image details necessary for effective anomaly detection. The Train data has 3066 good images. Validation data has 383 good images and 125 anomaly images. Test has 384 good and 127 anomaly respectively.

The preprocessing workflow began with defining a series of transformations applied to each image. We used the torchvision.transforms library to perform these operations, which included resizing, tensor conversion, and normalization. Each image was resized to 64x64 pixels to standardize input dimensions, making them compatible with the CAE model's architecture. The images were then converted into PyTorch tensors to be processed by the neural network. We applied normalization using a mean of 0.5 and a standard deviation of 0.5, which helped scale pixel values to the range of [-1, 1]. This normalization is crucial for stabilizing and speeding up the training process by ensuring that the network's inputs are on a common scale.

3.Model and Implementation

In this project,We can use different techniques like classifying or reconstructing to detct the anomaly. For reconstruction based methods, we use Convolutional Autoencoder (CAE) to tackle the anomaly detection problem using the MVTec AD dataset. The CAE is designed to compress the input images into a lower-dimensional representation and then reconstruct them, highlighting any deviations as anomalies. Below is a detailed breakdown of the model architecture and implementation process.

3.1 Model Architecture

The Convolutional Autoencoder model comprises two primary components: the Encoder and the Decoder.

Encoder:

The encoder reduces the spatial dimensions of the input image through a series of convolutional layers, effectively compressing the image into a lower-dimensional latent space.

We used several nn.Conv2d layers with a kernel size of 4,3,2 and stride of 2 or 1 to downsample the image progressively from 64x64 to 1x1.

Batch normalization (nn.BatchNorm2d) was applied after each convolutional layer to stabilize and accelerate the training process by normalizing the inputs to each layer.

LeakyReLU and ReLU activation functions (nn.ReLU) were used to introduce non-linearity, allowing the model to learn more complex patterns in the data. ReLU gave the optimal values compared to LeakyReLU with parameters of 0.1,0.2,0.3.

Decoder:

The decoder attempts to reconstruct the original image from the compressed feature representation created by the encoder.

We used nn.ConvTranspose2d layers to upsample the compressed feature map back to the original input size of 64x64.

Similar to the encoder, batch normalization and ReLU activation functions were employed in the decoder to enhance the learning process.

Finally, a Sigmoid activation function was used in the last layer matching the normalized input images.

The below diagrams represent encoder and decoder part respectively.

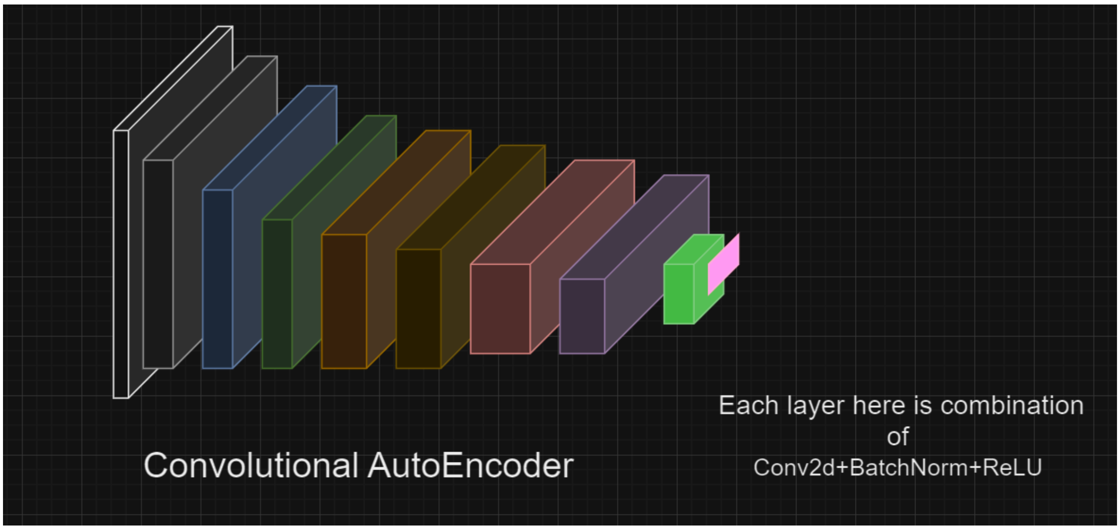

Fig 1 Encoder part of CAE

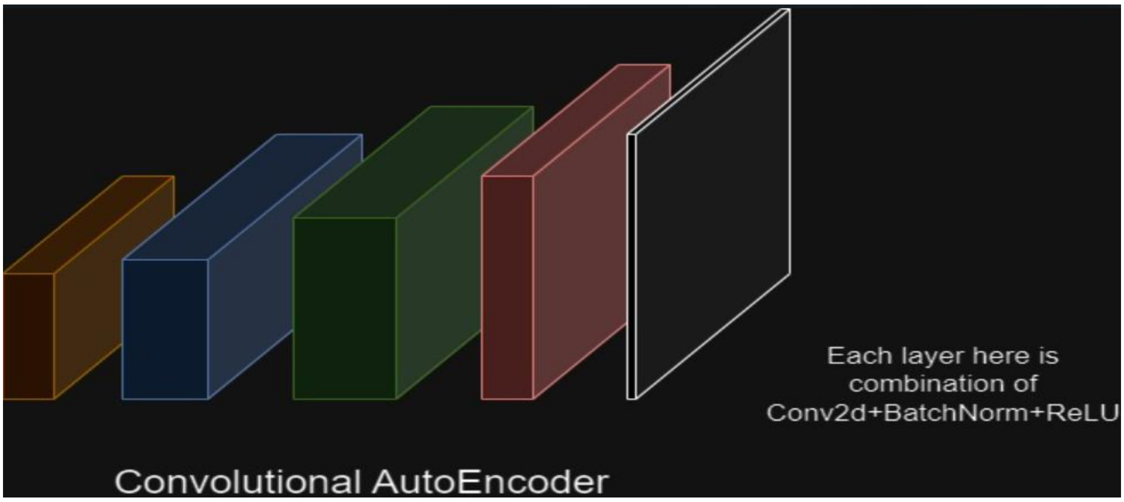

Fig 2 Decoder part of CAE

3.2 Implementation

We used the PyTorch framework for implementing this model.

We used 'torchvision.datasets.ImageFolder' to load the data. Using transformations library we preprocessed the data.

We employed a custom loss function that combines Mean Squared Error (MSE) and Structural Similarity Index Measure (SSIM) to evaluate the reconstruction quality. The SSIM loss emphasizes structural similarity, which is crucial for anomaly detection tasks.

We tried on different optimizers like Adam, SGD, RMS Prop. Adam gave the optimal results.

The model was trained for 25 epochs using the Adam optimizer with a learning rate of 0.001. The training loop involved the following steps:

- Forward pass: Input images were passed through the CAE to obtain reconstructed images.

- Loss computation: The loss between the input and reconstructed images was computed using the combined loss function.
- Backward pass: Gradients were calculated, and the model weights were updated using the optimizer.

The train and validation loss was plotted. The reconstruction errors for training images were computed and plotted in a histogram to visualize the distribution of reconstruction errors. Using it we can determine the threshold.

Using the parameters obtained in training, we tested our model on the test data which has both good and anomaly images. We used accuracy measure to measure how many prediction are correct.

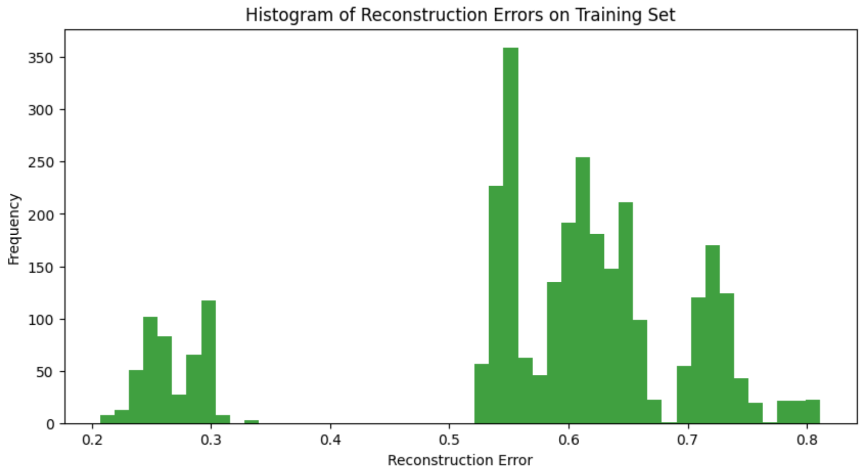

Fig 3 Training Reconstruction Errors

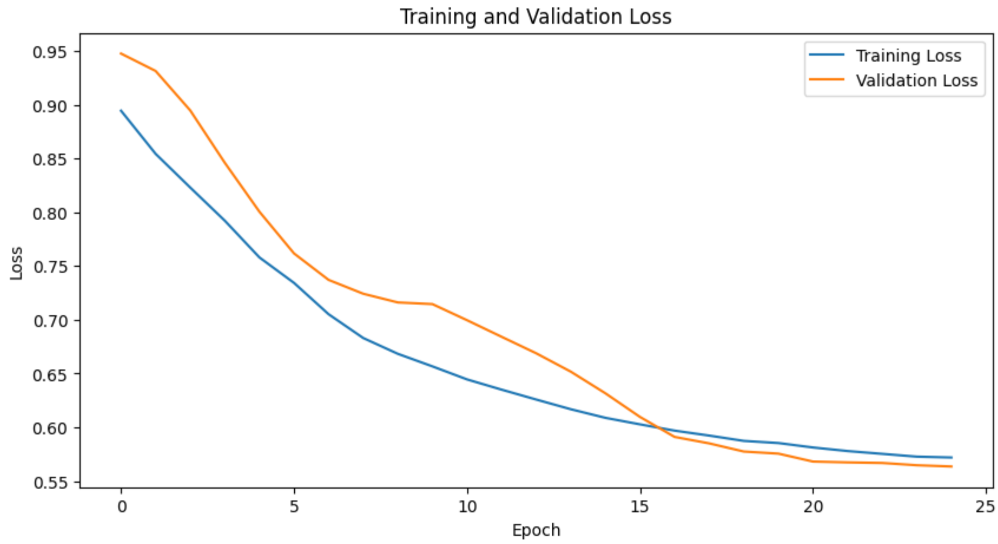

Fig 4 Train and Validation Loss.

4.Evaluation and Results

The Convolutional Autoencoder (CAE) model was evaluated on the MVTec AD dataset, achieving an accuracy of 77.1%. The following parameters were used during training:

- **Batch Size:** 1024
- **Optimizer:** Adam
- **Learning Rate:** 0.001
- **Alpha (Weight of MSE and SSIM):** 0.5

The model was trained for 25 epochs, utilizing a custom loss function that combines Mean Squared Error (MSE) and Structural Similarity Index (SSIM). The loss function aims to balance reconstruction accuracy and perceptual similarity between the input and output images.

The achieved accuracy demonstrates the effectiveness of the CAE model in detecting anomalies within the MVTec AD dataset. These results highlight the potential of leveraging convolutional autoencoders for anomaly detection tasks, especially in industrial settings where identifying subtle deviations from normalcy is crucial for ensuring product quality and reliability.

Conclusion:

In this project, we developed a Convolutional Autoencoder (CAE) model for anomaly detection using the MVTec AD dataset. The CAE was trainedwith the objective of reconstructing normal images accurately while highlighting anomalies in the reconstruction error.

The CAE architecture comprises an encoder and a decoder, with convolutional layers designed to capture spatial features of the input images. We utilized a custom loss function that combines Mean Squared Error (MSE) and Structural Similarity Index (SSIM) to train the model, aiming to balance reconstruction accuracy and perceptual similarity.

During training, the model achieved promising results, with the training and validation loss decreasing steadily over epochs. After 25 epochs of training, the CAE demonstrated an accuracy of 77.1% on the validation set. This accuracy indicates the model's ability to effectively differentiate between normal and anomalous images. Furthermore, the CAE was evaluated on a separate test set to assess its generalization performance.

We conclude that the developed CAE model shows potential for anomaly detection tasks, particularly in industrial settings where detecting subtle deviations from normalcy is essential for quality control and maintenance.

References

TensorFlow Authors. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Retrieved from https://www.tensorflow.org/

PyTorch Contributors. (2017). PyTorch: An open source deep learning platform. Retrieved from https://pytorch.org/

Paul Bergmann, Kilian Batzner, Michael Fauser, David Sattlegger, Carsten Steger: **The MVTec Anomaly Detection Dataset: A Comprehensive Real-World Dataset for Unsupervised Anomaly Detection**; in: *International Journal of Computer Vision* 129(4):1038-1059, 2021, DOI: 10.1007/s11263-020-01400-4.

Paul Bergmann, Michael Fauser, David Sattlegger, Carsten Steger: **MVTec AD — A Comprehensive Real-World Dataset for Unsupervised Anomaly Detection**; in: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 9584-9592, 2019, DOI: 10.1109/CVPR.2019.00982.

Hanxi Li, Jianfei Hu, Bo Li, Hao Chen, Yongbin Zheng, and Chunhua Shen. "Target before Shooting: Accurate Anomaly Detection and Localization under One Millisecond via Cascade Patch Retrieval."

Hyun, J., Kim, S., Jeon, G., Kim, S. H., Bae, K., & Kang, B. J. (Year). "ReConPatch: Contrastive Patch Representation Learning for Industrial Anomaly Detection."