

A descriptive study on development of a transfer learning based fault detection model using 2D CNN for air compressors

by

Moenuul Hasan

17301119

A.K.M. Faiyaz Uddin 17301088

Anoup Ghosh

17101518

A thesis submitted to the Department of Computer Science and Engineering in partial fulfillment of the requirements for the degree of B.Sc. in Computer Science

Department of Computer Science and Engineering
Brac University
June 2021

© 2021. Brac University
All rights reserved.

Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:



Moenul Hasan
17301119



A.K.M. Faiyaz Uddin
17301088



Anoup Ghosh
17101518

Approval

The thesis/project titled “A descriptive study on development of a transfer learning based fault detection model using 2D CNN for air compressors” submitted by

1.Moenul Hasan (17301119)

2.A.K.M. Faiyaz Uddin (17301088) 3.Anoup

Ghosh (17101518)

Of Spring, 2021 has been accepted as satisfactory in partial fulfillment of the re- quirement for the degree of B.Sc. in Computer Science on June 02, 2021.

Examining Committee:

Supervisor:
(Member)



Dr. Jia Uddin
Associate Professor
Department of Computer Science and Engineering
Brac University (On Leave)
Assistant Professor (Research Track)
Technology Studies Department
Woosong University, Daejeon, South Korea

Co Supervisor:
(Member)



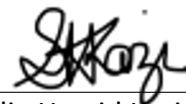
Rubayat Ahmed Khan
Lecturer
Department of Computer Science and Engineering
Brac University

Thesis Coordinator:
(Member)

Dr. Md. Golam Rabiul Alam
Associate Professor

Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)



Sadia Hamid Kazi
Chairperson and Associate Professor
Department of Computer Science and Engineering Brac
University

Ethics Statement (Optional)

We looked through a variety of articles and publications, as well as websites and discussion boards. The data has been taken from dataset repository (Verma et al., 2016). We have collected information from you videos as well.

Abstract

Fault Detection is essential for the safe and efficient operation of industrial manufacturing. Successful detection of fault features allows us to maintain a stable production line. Therefore, establishing a reliable and accurate fault detection method has become a huge priority now. Historically, various artificial intelligence-based models are used to predict faults in machines accurately to some extent. Mainly, machine learning and deep learning-based processes are being used. However, there are some shortcomings in those processes. Firstly, machine learning is mostly dependent on previous data and fails to recognize new issues that have not been introduced to the model during the training phase. Secondly, with deep learning, it is very time-consuming to reliably classify faults and difficult to establish an effective model for complex systems of current days. Thus, in our paper, we are proposing to use a transfer learning-based optimization of the deep learning process to meet the requirements of real-time fault classification and accurate detection of faults in adverse operational conditions. We will be using wavelet transformation of raw signal data to 2D images and constructing a DCNN based transfer learning architecture to extract the fault features of the machine. Finally, we will be feeding the network with data from our target domain for fine-tuning the network to work accurately in the target domain. We will be testing with two cases to find the accuracy and accuracy optimization over the deep learning (AAG) values of our system. Finally, we will be comparing our architecture with state-of-the-art transfer learning architectures from Keras.

Keywords: Fault Detection; Machine Learning; Deep Learning; Real-Time Fault Classification; Wavelet Transformation; 2D Image; DCNN; Keras

Dedication (Optional)

We'd want to dedicate our research to our cherished family, particularly our devoted parents. We would not be able to do our task successfully without their unwavering support and faith in us. We also dedicate our humble efforts to all dedicated and well-respected teachers.

Acknowledgement

First and foremost, we thank God for allowing us to finish our thesis without any serious setbacks. Then we wanted to express our gratitude to our supervisor, DR. Jia Uddin, for putting up with our blunders and offering constant feedback to help us improve our study. We'd like to express our gratitude to all of the helpful faculty members, as well as our co-supervisor, Mr. Rubayat Ahmed Khan. We also want to express our gratitude to our parents and teammates for their unwavering support throughout the semester.

Table of Contents

Declarationi Approvalii

Ethics Statementiv

Abstractv Dedicationvi

Acknowledgmentvii

Table of Contentsviii List

of Figuresxi List of

Tablesxiii

Nomenclaturexiv

1 Introduction1

1.1 Overview	1
1.2 Motivation	1
1.3 Problem Statement.....	2
1.4 Thesis Statement	2

2 Literature Review4

3 Background8

3.1 Air Compressor	8
3.2 Type of Air Compressors.....	8
3.3 Positive Displacement.....	9
3.3.1 Rotary Screw	9
3.3.2 Rotary Vane.....	9
3.3.3 Piston Type	10
3.3.4 Piston Type Air Compressor	10
3.4 Dynamic Displacement	11
3.4.1 Axial Air Compressor	11
3.4.2 Centrifugal Air Compressor	11
3.5 Single Stage Air Compressor	12
3.5.1 Components	13
3.6 Application of Air Compressor	14
3.6.1 Glass Manufacturing	14
3.6.2 Automotive.....	15
3.6.3 Nitrogen Plants.....	15
3.6.4 Chemical Manufacturing	15

3.6.5	Starting internal combustion engines	16
3.6.6	Electronics	16
3.6.7	Food and Beverage.....	16
3.6.8	Manufacturing.....	17
3.6.9	Aerospace.....	17
3.6.10	Hospital/Medical	18
3.6.11	Mining	18
3.6.12	Plastic Industries	19
3.6.13	Power Generation.....	19
3.6.14	Wood Products.....	19
3.6.15	Pharmaceuticals	19
3.6.16	Spray Painting.....	19
3.7	Machine Learning	20
3.7.1	KNN.....	20
3.7.2	Random Forest.....	22
4	Methodology	24
4.1	Workflow.....	24
4.2	Architecture Used	26
4.2.1	Lenet-5	26
4.2.2	Resnet50.....	27
5	Implementation of Proposed Models	29
5.1	Lenet-5 Based 2D Deep CNN	29
5.2	Implementation of Transfer Learning on our 2D CNN.....	35
5.3	Implementation of Transfer Learning Using Resnet50.....	35
6	Results	37
6.1	Model Summary Results From 2D CNN	37
6.1.1	Model Summary	37
6.1.2	Validation Accuracy Curve.....	38
6.1.3	Validation Loss Curve	38
6.1.4	Accuracy Curve	39
6.1.5	Testing Loss Curve	39
6.1.6	Confusion Matrix.....	40
6.1.7	Precision-Recall Calculations.....	40
6.2	Model Summary Results From Transfer Learning Framework	40
6.2.1	Model Summary	40
6.2.2	Validation Accuracy Curve.....	41
6.2.3	Validation Loss Curve	42
6.2.4	Accuracy Curve	42
6.2.5	Testing Loss Curve	43
6.2.6	Confusion Matrix.....	43
6.2.7	Precision-Recall Calculations.....	44
6.3	Model Summary and Results from Transfer Learning using Resnet50	44
6.3.1	Model Summary	44
6.3.2	Validation Accuracy Curve.....	45
6.3.3	Validation Loss Curve.....	45
6.3.4	Accuracy Curve	46

6.3.5	Testing Loss Curve	46
7	Result Analysis	47
7.1	Analysis of Results from 2D CNN	47
7.2	Analysis of Results from Transfer Learning Framework.....	47
7.3	Analysis of Results from Resnet-50 based Transfer Learning	48
7.4	Comparison Between 2D CNN based transfer learning and Resnet-50 based Transfer Learning.....	48
8	Discussion and Future Work	49
9	Conclusion	50
	Bibliography	54

List of Figures

3.1	Positive Displacement.....	9
3.2	Piston Type Air Compressor.....	10
3.3	Dynamic Displacement	11
3.4	Suction of Air.....	12
3.5	Compression of Air	12
3.6	Working Principle of Single Stage Air Compressor.....	13
3.7	Components of Single Stage Air Compressor	14
3.8	Machine Learning Workflow	20
3.9	KNN Workflow	21
3.10	KNN Classification	22
3.11	Random Forest - Flow Diagram.....	22
3.12	Random forest sample with replacing.....	23
4.1	Workflow Diagram	24
4.2	Sample Data	25
4.3	Data Splitting.....	25
4.4	Internal Structure of Lenet-5 Architecture	26
4.5	Each Layers of Resnet50 Architecture.....	28
5.1	First Convolution Model	29
5.2	Creating First Convolution Model.....	29
5.3	First Subsampling Layer	30
5.4	Creating First Subsampling Layer	30
5.5	Second Convolution Model.....	30
5.6	Creating Second Convolution Model	31
5.7	Second Subsampling Layer	31
5.8	Creating Second Subsampling Layer	31
5.9	Second Subsampling Layer	32
5.10	Creating Second Subsampling Layer	32
5.11	Image of Full Model	33
5.12	Implementation of transfer flow	35
5.13	Importing Resnet50	35
5.14	Our Layer	35
6.1	Validation Accuracy	38
6.2	Validation Loss	38
6.3	Accuracy Rate Graph.....	39
6.4	Testing Loss	39
6.5	Confusion Matrix	40

6.6	Validation Accuracy Graph	41
6.7	Validation Loss Graph	42
6.8	Accuracy Graph.....	42
6.9	Testing Loss	43
6.10	Confusion Matrix	43
6.11	Validation Accuracy (Resnet50).....	45
6.12	Validation Loss (Resnet50).....	45
6.13	Accuracy Graph (Resnet50)	46
6.14	Testing Loss (Resnet50).....	46

List of Tables

6.1	2D CNN model	37
6.2	Precision-Recall Table.....	40
6.3	TCNN Framework	41
6.4	Precision-Recall.....	44
6.5	Model Summary of Resnet50	44

Nomenclature

The next list describes several symbols & abbreviation that will be later used within the body of the document

ASI Acoustic Spectral Imaging

BDC Bottom Day Center

CNN Convolutional neural network *DCNN*

Deep Convolutional neural network *DNS*

Domain Name System

GLCM Gray Level Co-occurrence Matrix

LBP Local Binary Pattern

LIV Leakage Inlet Valve

LOV Leakage Outlet Valve

ML Machine Learning

NRV Non-Return Valve

OAA Open Agent Architecture

PCA Prompt Corrective Action

PMF Probability Mass Function

RP Recurrence Plot

SIFT Scale-Invariant Feature Transform

STFA – PD Sparse Time-Frequency Analysis - Primal-Dual

SURF Speeded-Up Robust Features

SV M Support Vector Machine

TCNN Temporal Convolutional neural network

TDC Top Day Center

TFI Time-Frequency Image

Chapter 1

Introduction

1.1 Overview

Due to the never-ending growth in global population and demand, industrial manufacturing has become the biggest issue of the 21st century. The manufacturing processes are mostly semi or fully automated via the use of machines nowadays. The efficiency of the process alongside the safety of the personnel involved in the process is mainly dependent on the operational accuracy of said machines [13]. One of the commonly used machines in the industry is the industrial air compressors. From the air compressor, the power we get can be used to replace steam and electricity in most engineering and manufacturing areas. Compressed air helps to save time, money, and physical strain. There are many industrial uses of air compressor such as, glass manufacturing, automotive, nitrogen plants, chemical manufacturing, starting internal combustion engines, electronics, food and beverage, general manufacturing, aerospace, hospitals/medical, mining, plastics, power generation, wood products, pharmaceuticals, refrigeration, oxygen plants, spray paint, among other applications. Compressed air is a valuable resource since it can be stored and used as needed. The cost of production is lower than other forms of energy because it is a tidy and clean source of energy. Compressed air is used in almost every industry. To ensure the safety and efficiency of the air compressor, we need to detect faults of the air compressor as early as possible so that they can be fixed early[12].

1.2 Motivation

Among the methods used to monitor the accuracy and predict further occurrences of operational errors, industrial fault detection has been the most effective [9]. Fault detection monitors the operational behaviors of a machine, evaluates its effectiveness, and classifies the behaviors as effective or faulty [8]. Traditionally, three techniques are used in industrial fault detection which includes model-driven, experience-driven, and data-driven fault analysis. The data-driven method uses the previous operational data to learn about fault modes without establishing a precise model of the systems. Generally, machine learning-based algorithms such as KNN, SVM, etc. are used to develop models for classifying faults in industrial machines [19]. Nowadays, due to the increase in computational power in consumer computers, hardware-accelerated deep learning models like TensorFlow, Keras, etc. are also used in fault classification model development. These models are easy to use but they have their

shortcomings as well. Machine learning models are highly reliant on previous data and fail to detect any new faults. Moreover, this method of fault detection can only function within the same domain of data. If the training and validation domain is different from the test domain, machine learning-based models cannot function. On the other hand, deep learning is much more effective in this regard. Deep learning-based neural networks are far more superior and complicated architectures capable of being domain-independent and much more efficient in classifying the data. But generating a neural network for a huge task such as industrial fault diagnosis can be very tricky. As the size of the data grows, the architecture of the model complicates exponentially. It is the most helpful while evaluating a very complicated system as it is really hard to establish an explicit model of such systems.

Now in this paper, we are trying to eradicate the aforementioned shortcomings of both machine and deep learning in data-driven fault diagnosis. We propose to use transfer learning on a previously trained deep convolutional neural network (CNN) that has been trained with similar data and establish a model to accurately classify faults in complex machines. We are considering converting the raw sensor data to 2D images using the wavelet functions. Then we are going to create a CNN-based architecture using the images. Finally, the CNN will be fine-tuned using data from our target domain.

1.3 Problem Statement

One of the most commonly used tools in data-driven fault detection is machine learning [19]. Machine learning-based fault diagnosis is conducted using two steps which are collection and selection of the fault features and classification of fault types. Even though historically machine learning has been quite effective in accurately detecting and predicting faults, there are a few downsides to machine learning in terms of the most optimized fault detection. Machine learning is very reliant on the experts' previous knowledge while extracting the features. Moreover, machine learning is most effective when designed to perform specified tasks and fails when trying to achieve a generalized performance for future problems.

1.4 Thesis Statement

To remedy some of the issues that occur in machine learning, deep learning is being implemented by many experts nowadays. Deep learning allows the model to forgo its reliability on the experts as it can automatically extract the data needed to select features from live signals. This eliminates the chances of experts' influence over the dataset. However, there are still some shortcomings present in deep learning. Due to the huge computational complexity of deep learning, it is often unreasonable to use it. Especially, considering the complex nature of the systems as well. Additionally, deep learning assumes the target domain as the same as its training domain, reducing its overall effectiveness while trying to fit into the target domain. To solve some of the issues that occur in deep learning, we are proposing to detect fault using the transfer learning technique. In transfer learning, one pre-trained model's weights are reused to optimize the performance of another model from the same or similar target domain. This approach allows for faster or more accurate classification when

modeling the second task. In transfer learning, pre-trained weights and features can be transferred to the new target network to be trained on a target dataset. Thus, it reduces the computational complexity significantly.

Chapter 2

Literature Review

In order to gain additional information regarding our topic, we have gone through some of the previously published papers on this topic. These papers provided us with a thorough picture of the current landscape of the field alongside showing us the shortcomings of previous works. We have tried to find out the opportunities to work on further improvements in Fault Detection.

Among the papers that we have read, almost all of them talk about the inefficiencies of data-driven fault diagnosis using machine learning or deep learning. This information helped us in finding out our preferred method of fault diagnosis which is Transfer Learning. To generate a feasible model of unsupervised fault detection, we had to learn how to extract real-time data from the machines. From these papers, we have learned how we can transform the raw signal data from the machine-mounted sensors into 2D image data for our DCNN to work on for feature extraction. Additionally, we have learned how to use the LeNet-5 architecture to build a DCNN for using it as an input CNN of our TCNN framework.

This paper studied an approach for faulty severity monitoring of rolling bearings based on the texture feature extraction of sparse time-frequency images (TFIs) [16]. This study, suggests the novel one, a new time-frequency analysis method, named STFA-PD. Then Gray level co-occurrence matrix (GLCM)-based texture features of the sparse TFIs were extracted to realize fault severity monitoring. STFA-PD method can overcome the drawbacks of other methods. Therefore, It can not only used to obtain high-quality TFIs but also so universal that it can be used for other things. So, this study showed that this method has significant potential to be a powerful tool for the fault severity monitoring of rolling bearings. There are also some limitations of this model. Firstly, a training feature set is needed for this approach. So, it might be difficult to obtain in a practical situation. Secondly, the GLCM-based features are not correlated with physical significance. So, it might be hard for us to identify the bearing fault severity from the feature values.

This paper includes the Local Binary Patterns (LBPs) to find out the problem and get the solution for the problems [7]. It is a two-dimensional texture analysis. This paper used the case of eight different motor operating situations, and great diagnostic performance was obtained. It changes the signal (1D) to image (2D) then extracts the feature descriptors. If the problem can be found, then classify the problem, and if the model can't find the problem, then train it and find the problem again. This paper shows what problems cause the different vibration signals and corresponding gray-scale images, so the problem can be found by looking at the

vibration signals and gray-scale images. The common limitation of the LBP is that 3x3 matrix block cannot capture the dominant features if the structure's scale is large. Also, this paper only includes 8 different situations. It does not have enough experiments. To find out more motor faults, the proposed model should try to find out more problematic motors and find out the solution.

The authors of the research used the model called 'Fault Feature Extraction in order to find out shortcomings of Gear [6]. The researchers used 'envelope extraction' methods including Hilbert Transform demodulation, detector-filtering method, and high-pass absolute value demodulation method. The main tool in distributing time- frequency signals was S-Transformation. In the analysis of the findings, all features combined in GLCMs provide statistics. Although the findings are highly reliable in terms of practicality, some factors are not considered in the research. Some features do not suit in GLCMs and not appear in statistics. Moreover, S-Transformation underestimated the tiny signals in distribution.

The proposed model uses image feature extraction, which is a recurrence plot (RP) that shows time-related information in the global topological properties of the system [21]. That is, when the system is stable, the texture of RP is evenly distributed, and when the system is unstable the RP texture will show relevant information. SIFT algorithm is needed for an automatic feature point extraction. The main limitation of the model is the inability to store key point descriptors of an image rather than an original image in order to save space. What is more, images cannot be smaller sized than the original image.

Based on acoustic spectrum imaging of acoustic emission signals, this research provides a viable problem diagnostic methodology [18]. ASI provides a visible representation of acoustic emission spectral features in images. The proposed measurement provides a robust classifier technique with high diagnostic accuracy. This paper presents two major limitations. First, the requirement of domain-level expertise for feature extraction and selection under different operational speeds, and second is that the requirement of special dynamic algorithms for automation of the feature extraction process.

This paper introduces a method for identifying problems that occur with Induction motors during their usage, using many different ways to detect failure [11]. The most used being Vibration analysis. The proposed model is divided into 4 parts. They first do data conversion, which converts vibration to produce 2D gray-level images. Then generate a DNS map to extract texture features and select the most distinctive figures using PCA. Lastly, utilizing SVM to classify the data and make a final decision to determine what kind of problem there is.

In this paper, the authors proposed a new method for fault diagnosis of rolling bearings based on the SURF algorithm, where the two-dimension signal is used [34]. It is different from other classical 1-d processed methods, it transforms 1- dimensional images into signals. SURF is a computer vision technique that improves the SIFT methodology by extracting local information more efficiently using picture texture. SURF is developed from the SIFT algorithm, and it is a novel detector- descriptor scheme. Efforts have been made to develop a robust fault diagnosis system for rolling bearing. The model approach translates the input vibration signals into gray-scale images for the fault classification of the rolling bearing. The SURF-based algorithm is presented, for the sake of extracting texture features from the images. The proposed model method refers to rolling bearings and effectively classifies each

vibration signal to its corresponding fault category. The limitations of the model are the cost of the high computation for processing Of 128-dimensional descriptors. Also, another one is that the number of feature descriptors for each image is uncertain, sometimes the number may even be too small to classify the type of faults.

This study describes a two-dimensional Shannon wavelet representation for extremely accurate fault identification of numerous induction motor problems [10]. In this study, a well-known image feature extraction scheme is employed, which was proposed by F. M. Khellah et al. to generate more effective and robust features of the induction motor. SVM is designed to separate the data into one of two classes and the OAA method is employed, which is one of the most popular and simplest techniques for multi-class classifiers. In order to detect and identify bearing defects in the induction motor, Algorithms 1 and 5 utilized time-domain features from vibration signals such as statistical features (e.g., root mean square, variance, and skewness) for a low computational burden, and zero-crossing features, which are durations between the successive zero-crossing intervals. Some of the weaknesses of the Shannon-Weaver Model The transmission model does not consider the medium that is used. The channel might be noisy, and the receiver might not be able to decode it, causing complications in the communication process.

This paper explores an imaging-based approach to achieve rotational speed independence in bearings by using vibration signal imaging and local binary patterns [14]. They use 16 datasets using one classifier for the entire dataset. Case Western Reserve University's Bearing Data Center puts the suggested method to the test using seeded fault test data (2016). On the rollers, inner raceways, and outer raceways of the test bearings, single-point localized flaws were seeded. The classification accuracies for all fault conditions more than 95

In this paper, the network learns from the massive source data set and that knowledge is applied to targets data to identify faults [17]. It has three methodologies that are source task, element transfer, and target task. They have three common they are data, vibration image, and convolutional neural networks. It works on mainly transform vibration which works is based on collecting the data and the collected data can be check so that we do not have to load time and again. Transfer learning allows an established model to identify errors that emerge under different working situations by using feature information obtained under one set of settings through hidden layers. To make the automated feature extraction more reliable and accurate, the discrete orthonormal Stock well transforms (DOST) was proposed as a preprocessing step for creating a load- and rpm-invariant scenario for considering signals of multiple health types. Experimental results showed that the proposed method achieves an average of 99.8

After that, we have looked for all the shortcomings in the papers to come up with the aspects of the field of study where we can bring improvements with our study. Firstly, in paper 1, we have learned that we should not use very low-level data in our input CNN. Moreover, using any kind of parallel computation method such as spark is more feasible to reduce the time complexity of the TCNN framework. Secondly, in paper 2, we have learned that by using only TCNN on supervised fault detection, we will only be able to detect one type of fault emerging in the future. Thirdly, in paper 3, we have learned deeply about the mechanisms of deep learning in fault detection and how it is failing in executing the real-time classification of faults due to its high computational time complexity. Furthermore, in paper 4, the methods they

used were similar to one of the problems with machine learning. This was previous supervision of labels. Moreover, the paper taught us to properly study the various layers of the model for proper performance optimization. Finally, from paper 7, we have learned how we can apply our model for fault detection in different working conditions to calculate a much more representative result that is as close as possible to real-life scenarios.

The aforementioned words were some of the many useful information that we had gathered from the papers that we studied for our work. This was really helpful for us in painting a clear picture of the problem at hand and also in finding a straight path that we can take to reach our objective precisely and effectively.

Chapter 3

Background

3.1 Air Compressor

Compressed air is provided by air compressors in a variety of industries for a variety of applications. Air compressors are now used to power construction and manufacturing equipment, as well as to control system valves; earlier compressors were much less versatile. Air compressors have been around for thousands of years. In the 18th century, air compressors were used for something other than metalworking; they were also used for mining, fabricating metals, and ventilating underground areas. Air compressors were frequently used to carry heavy air quantities into the 8-mile construction tunnel during the 1857 construction of the Italy-France rail system. People soon came up with new ideas for how to use the technology. People started utilizing air compressors to convey energy around 1800. In 1888, Austrian engineer Viktor Popp built the first compressor plant in Paris, which grew from 1,500 kW to 18,000 kW in just three years. More air compression innovations improved the process, and it was soon incorporating electricity and pneumatic energy. There are so many different types of modern air compressors to choose from today. Depending on unique needs, Compressed Air Systems provides a diverse product line that includes reciprocating, oil-less, vehicle-mounted, and other air compressors.

Air compressors are vital parts of every industry or workshop. In recent years, they've shrunk in size and heft, making them more adaptable to a variety of work environments. These are compact units that provide energy to single air tools. Air compressors pressurize containers by pumping air into them. The air is then driven through a hole in the tank, causing pressure to build up. Consider it as an open balloon: once the compressed air is released, it may be used as energy. An engine converts electrical energy into kinetic energy, which powers air compressors. A crankshaft, piston, valve, head, and connecting rod are all used in the same way as in a combustion engine. The compressed air may then be utilized to power a wide range of tools. Nailers, impact wrenches, sanders, and paint sprayers are some of the more common alternatives.

3.2 Type of Air Compressors

Positive and dynamic displacement are the two strategies for creating air compression [25]. Each approach has multiple sub-categories, which we'll go over in more

detail later. Although the results are identical, the methods used to reach them differ.

3.3 Positive Displacement

In a positive displacement air compressors compress the air by forcing it into a chamber with a smaller capacity. Positive displacement refers to a variety of air compressors that are powered by positive air displacement. While the internal systems of different machines change, the mechanism of supplying electricity is the same. Some positive displacement compressors are more suited to industrial applications, while others are better suited to amateurs or personal projects (BigRentz, 2020) [25].

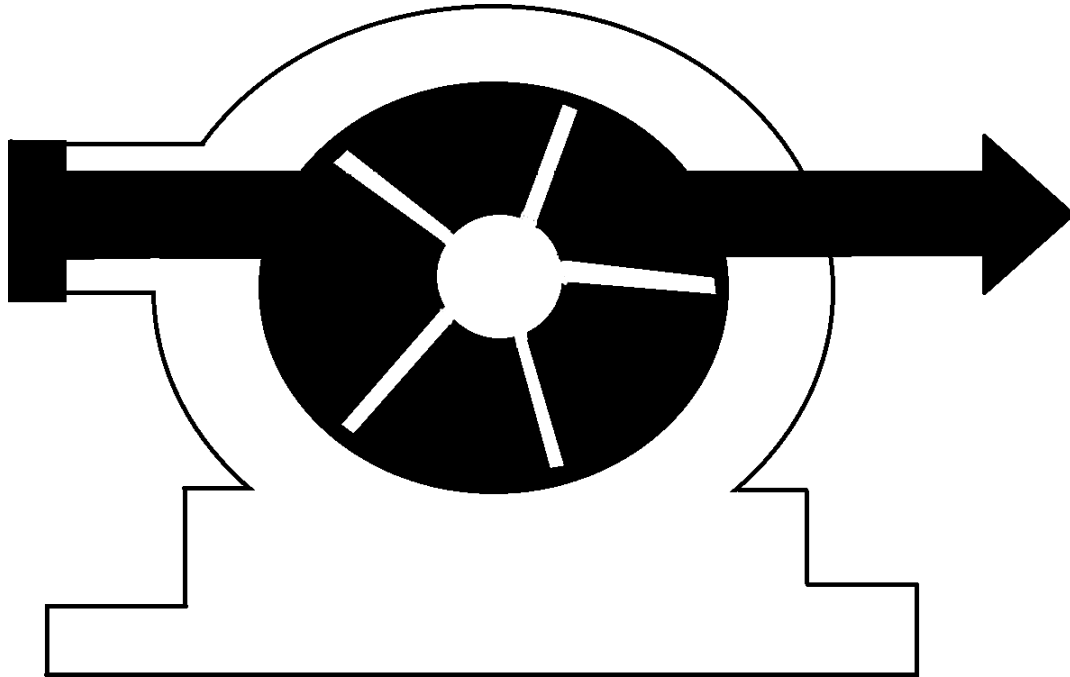


Figure 3.1: Positive Displacement

There are three basic types of positive displacement air compressors. Those are rotary screw, rotary vane, and piston type.

3.3.1 Rotary Screw

In a rotary screw air compressor, two internal screws are rotating in opposite directions. Inside these opposite rotating screws, it tapped and compressed the air. As they revolve around, the two screws provide continual movement. This type of air compressor is very easy to maintain. It is used for industrial applications and ideal for long-term operations (BigRentz, 2020) [25].

3.3.2 Rotary Vane

n rotary vane air compressors, vanes are placed on a rotor and spun inside the chamber instead of a screw. The air is compressed between the vane and its case

before being expelled through a separate exhaust outlet. As it is very easy to use, so it's widely used in personal projects.

3.3.3 Piston Type

To deliver gas at high pressure, a piston-type air compressor employs pistons operated by a crankshaft. It is used for smaller projects. There are two types of piston-type air compressors. Single-stage and two-stage air compressor (BigRentz, 2020) [25].

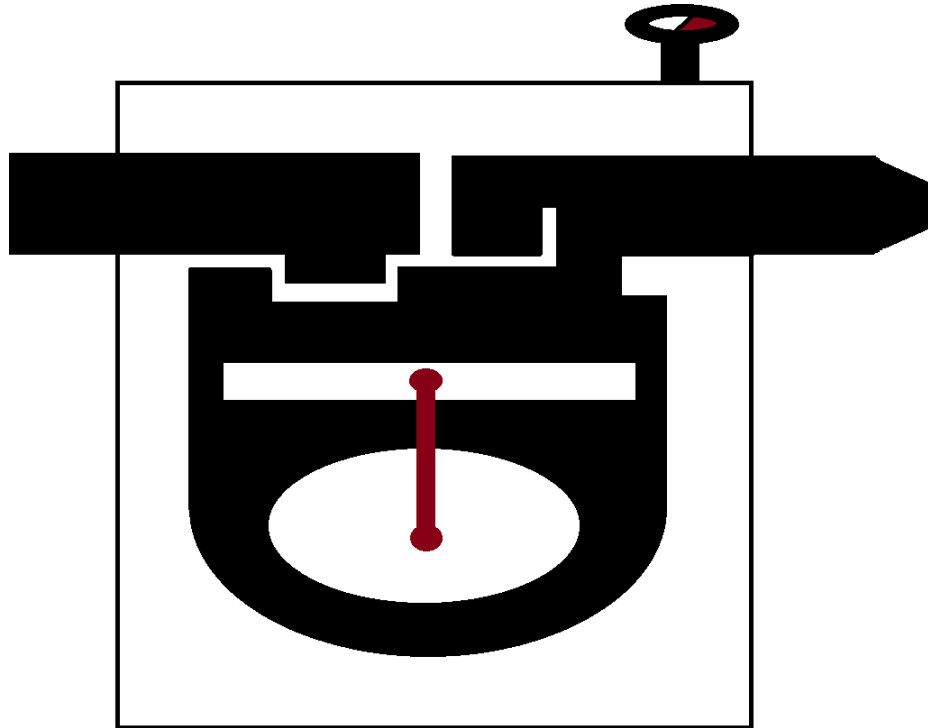


Figure 3.2: Piston Type Air Compressor

3.3.4 Piston Type Air Compressor

Single Stage

Two cylinders or air storage chambers are included in the single-stage air compressor. The first piston compresses air and delivers it into the second storage tank to keep the compressed air for future use. Electricity or gas are used to operate a single-stage air compressor (BigRentz, 2020) [25].

Two Stage

Two-stage air compressor consists of two compression chambers. A constant trickle of water across the engine is commonly used to cool double-acting compressors. Only a two-stage air compressor has this cooling system. Two-stage compressors are preferable for factories and workshops than private projects due to their high cost (BigRentz, 2020) [25].

3.4 Dynamic Displacement

2.2.2 Dynamic Displacement: To create airflow, dynamic displacement compressors use a revolving blade driven by an engine. After that, the air is constrained to generate pressure and within the compressor, the kinetic energy is stored. These air compressors are used in industrial applications such as chemical plant, steel manufacturing industries, glass manufacturing industries, etc. as it is designed for large industries.

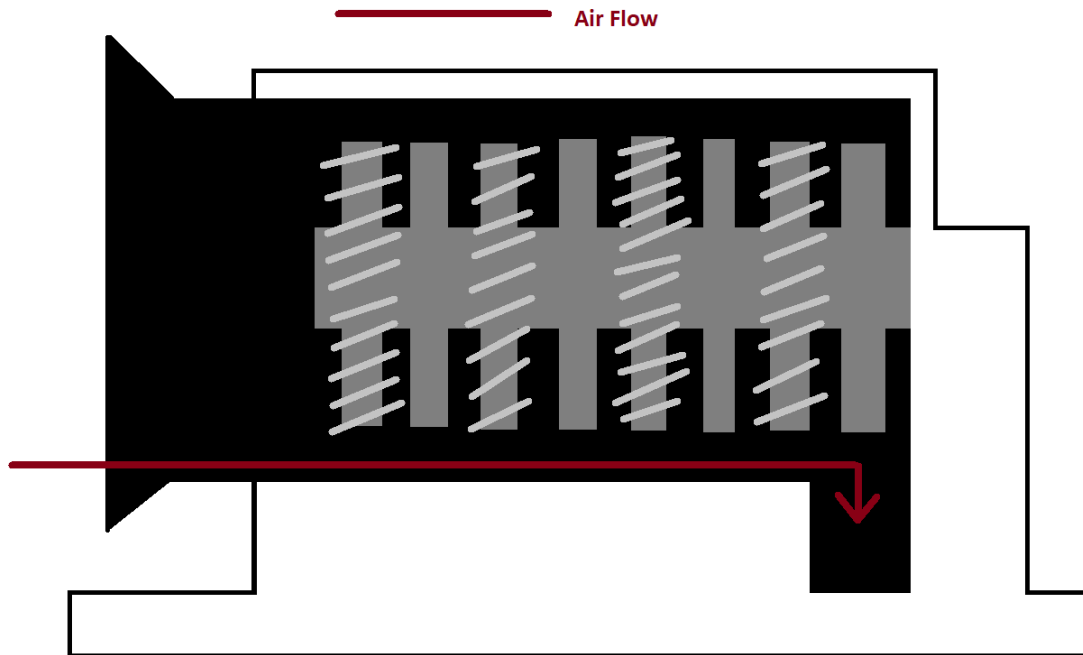


Figure 3.3: Dynamic Displacement

Dynamic displacement is divided into two categories, similar to positive displacement compressors: axial and centrifugal (BigRentz, 2020) [25].

3.4.1 Axial Air Compressor

Axial compressors create air by pushing it through a narrow space using a set of turbine blades. Axial compressors, like other bladed compressors, function with stationary blades that decrease airflow and increase pressure. This type of air compressor has limited uses. As they are mainly used in aircraft engine manufacturing industries.

3.4.2 Centrifugal Air Compressor

In a centrifugal compressor, the air is drawn into the center of it by a revolving impeller, which is then pushed forward by centrifugal force. More kinetic energy is created by slowing the flow of air via a diffuser (BigRentz, 2020) [25].

3.5 Single Stage Air Compressor

A single-stage reciprocating air compressor was used to collect data that we work with. A piston and a pressure-sensitive valve are used to operate a single-stage air compressor [27].

It's built around a single cylinder that compresses air with a single-piston stroke. For generating the necessary force to compress the air that cylinder is attached to a power source. A single-stage air compressor has only one cylinder and valve which makes it different from the two-stage and multi-stage air compressor. The pressure measurement valve and the piston push the single-stage piston compressor. The piston is used in the compression operation. The piston is controlled by a connecting rod and a crankshaft. To control the torque produced by the crankshaft's movement, appropriate flywheels are installed. The crankshaft of these compressors is powered by an electric motor.

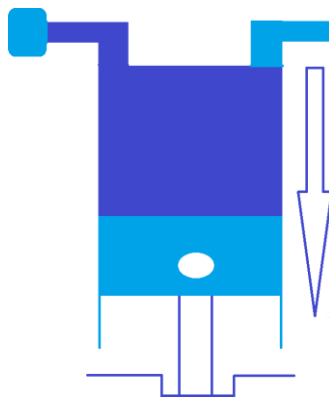


Figure 3.4: Suction of Air

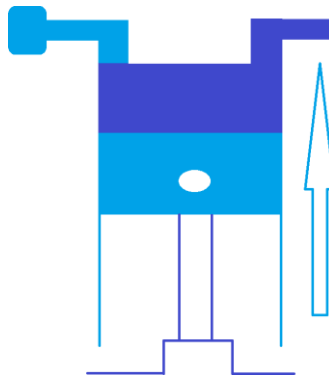


Figure 3.5: Compression of Air

All of the compression happens in a single cylinder in a single-stage reciprocating compressor. The cylinder is connected to two valves: an inlet or suction valve, and an exit or delivery valve. The friction differential affects how a spring or plate valve opens and closes. Cams regulate the operation of mechanical valves used for suction and discharge. As a result, when the piston is at TDC, the amount of air in the cylinder is negative. In this situation, clearance volume should be ignored.

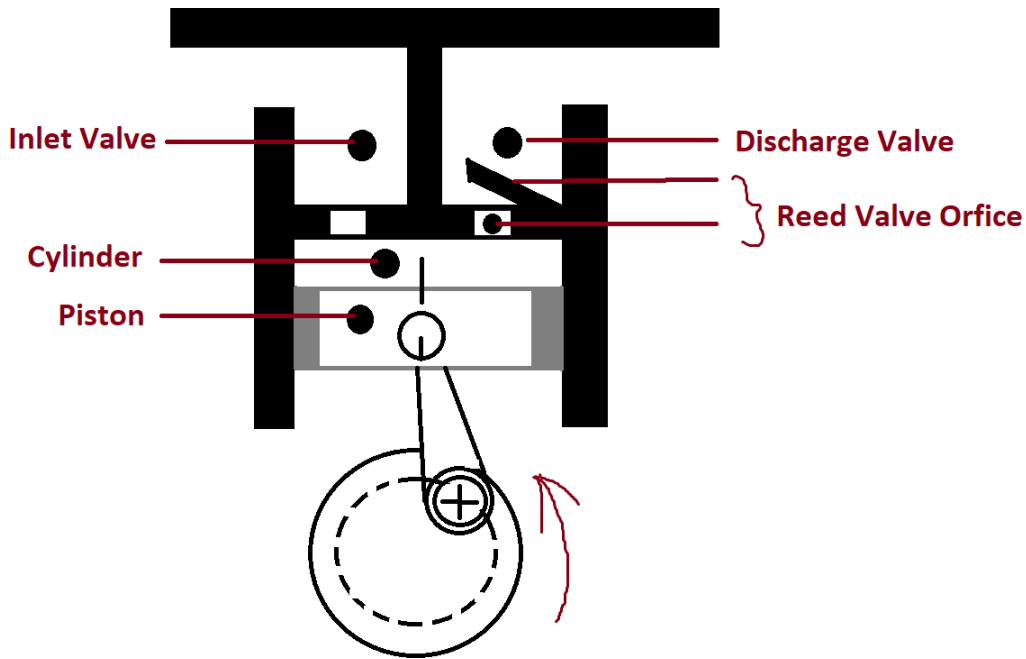


Figure 3.6: Working Principle of Single Stage Air Compressor

The cylinder pressure decreases below ambient pressure as the piston pushes downward. The suction valve is opened as a result of the pressure differential. The suction valve allows air to penetrate the cylinder at this stage. This is referred to as a "suction stroke". Due to crankcase movement, the piston moves upward and compresses the air as it passes BDC. At the compression phase in the cylinder, the internal cylinder pressure reaches a point where it is slightly higher than ambient pressure, at which point the suction valve shuts. The receiver is connected to the pressure valve. The distribution or outlet valve opens when the compressed air pressure reaches the pressure at the receiving end, and compressed air is released. As a result, it's known as "The Delivery Stroke." This is a single-stage air compressor compression stroke. The discharge valve opens at the end of this stroke, and compressed air is delivered to the receiver. The piston spins at a high rate, and the energy applied to the cylinder by the piston is enormous. It shortens the life of the compressor. A minor curvature at the tip of the cylinder prevents this.

3.5.1 Components

Piston

This section of the compressor is used to compress air. In the cylinder, it goes forward and backward. As it goes backward, it suckers air, and when it moves forward, it compresses air.

Crankshaft

For air compression, it rotates the piston. The electric motor is directly connected to the crankshaft

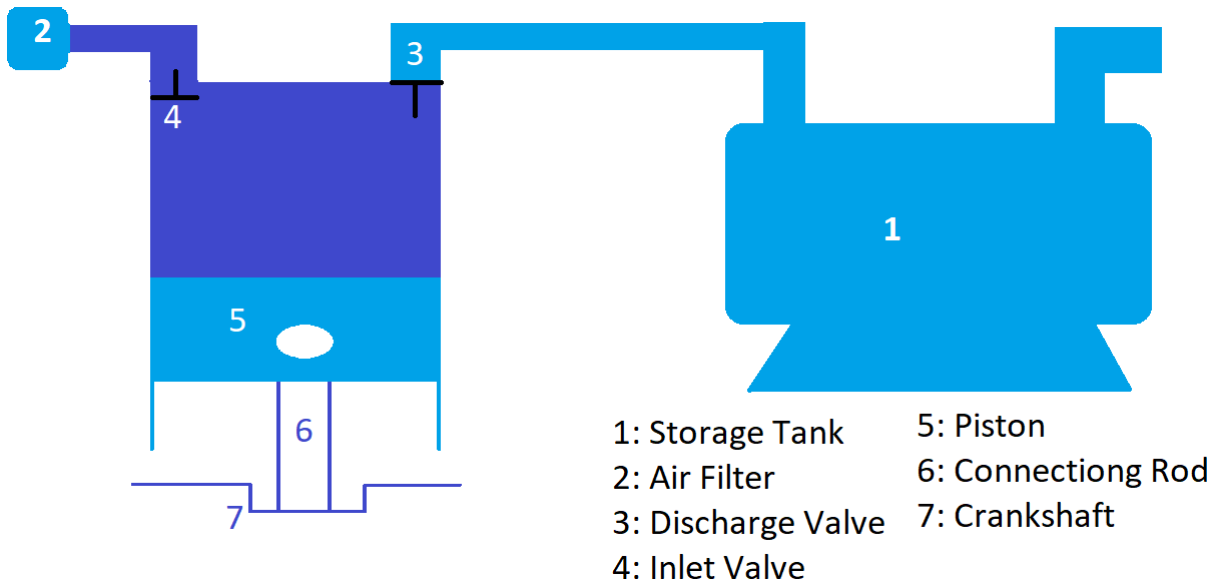


Figure 3.7: Components of Single Stage Air Compressor

Inlet Valve and Discharge Valve

The suction valve is also known as the inlet valve. It aims to draw air into the compression chamber. When the internal cylinder pressure drops below the external pressure as the piston pushes downward, the suction valve opens. The discharge valve, on the other hand, is used to discharge compressed air into a receiver or holding tank. If the internal cylinder pressure exceeds the external pressure, this valve opens. The outlet or distribution valve is another name for the discharge valve. The cylinder is directly connected to these single-stage compressor elements. Cams monitor the operation of inlet and outlet valves.

Connecting Rod

The piston movement is regulated by this compressor feature. The piston and the crankshaft are linked by the connecting rod.

3.6 Application of Air Compressor

3.6.1 Glass Manufacturing

Air compression system is an important part of bottle making plant and glass sheet manufacturing factory. This system helps to move everything from the silos to the finishing phases [29]. Air compressors are also used in heavy machinery that carries tons of raw materials. These materials are transported to the mixing chamber and forming phase, which create one-of-a-kind bottles and glass slabs. The use of compressed air and gas in the glass factory's smelting furnace aids in achieving a high burn-away rate. Because of friction, high-pressure compressed air can drill through the glass. Sandblasting is used to etch the pattern on the glass. By using compressed air to cool some glass sheets to a high temperature, the glasses become

breakage-proof and have a higher tensile strength. To reduce the risk of breakage, the glass sheets are lifted by vacuum chuck.

3.6.2 Automotive

Many people are familiar with the use of compressed air to inflate tires. While air compression for body shops and vehicle technicians is a popular mainstream and professional function, it extends well beyond correctly inflated tires. Air compressors will be used in auto body businesses, particularly for repainting vehicles. However, these companies and their staff do not stop there [30]. Nozzles and couplers, dryers, brushes, blowguns, and lubricators are all examples of this equipment. These products are portable and do not require batteries. They come in a variety of sizes, don't require a lot of storage space, and are convenient to educate technicians on. These air compressor pneumatic tools — and more — will be found in each functioning garage with an auto-body reputation to improve the pace and quality of their job.

3.6.3 Nitrogen Plants

Nitrogen is widely used in industrial sectors including perishable food packaging in the atmosphere [24]. It's also used to keep chemical plants safe from fires and explosions. Providing the generators with high-quality compressed air ensures long, trouble-free service and optimal performance. Air compressors offer a complete line of nitrogen generating systems that contain everything you'll need to get started. To provide the best quality air supply for the generators, air compressors and pre-treatment kits incorporate adsorption dryers and coalescing filters. With minimal additional floor space, the gas may be produced from your current system. High-pressure cylinders, liquid micro tanks, and bulk storage containers are common gas delivery techniques. Employees are safer in the workplace with an air compressor generator, which eliminates the dangers associated with older techniques. When compared to outsourced nitrogen sources, on-site nitrogen production provides more flexibility and saves time and money.

3.6.4 Chemical Manufacturing

The chemical sector places extremely high demands on equipment. Air compressed comes into touch with the process it supports in many applications, thus its quality is crucial. Compressed air is commonly used in the chemical industry for the following applications:

- Cleaning, aeration, and product movement all need process air, which is utilized in direct contact with the product.
- Compressed air controls regulate valves and cylinders, which are employed in the production process.
- Material handling — air-operated fluid pumping devices are employed in potentially explosive settings.
- Air is filtered via a membrane to create nitrogen, which is employed in several chemical applications.

- Air curtains used to create a secure and sanitary environment Drying of products – air is combined with the product to speed up the drying process.

3.6.5 Starting internal combustion engines

A compressor for charging an internal combustion engine consists of a compressor that blows compressed air into the engine's intake manifold and a fuel-powered turbine that drives the compressor [31]. The ventilation from a small gas-powered turbine is combined with the driving turbine of a regular turbocharger in one embodiment. The drive shaft of a small gas-powered turbine is combined with the drive shaft of a regular supercharger in a second embodiment. A compressor turbine with an air intake compressed air outlet, and bleed air outlet is combined to a small gas-powered turbine in a third embodiment, with the bleed air outlet supplying the gas turbine's combustor intake. The compressor is driven by a gas turbine, and compressed air is received from the compressor via the bleed outlet. The compressor's compressed air outlet can provide a constant boost, requires no engine horsepower, is simple to install, and does not require coupling to a rotating shaft or the engine's exhaust system. A compressor turbine with an air intake compressed air outlet, and bleed air outlet is coupled to a small gas-powered turbine in a third embodiment, with the bleed air outlet supplying the gas turbine's combustor intake. The compressor is driven by a gas turbine, which also receives compressed air via the bleed outlet. The compressor's compressed air outlet can provide a constant boost, requires no engine horsepower to operate, is simple to install, and does not require coupling to a rotating shaft or the engine's exhaust system.

3.6.6 Electronics

Compressors of air are commonly used to power a wide range of electrical and electronic devices [23]. To provide the power required to actuate pneumatic devices or perform cleaning operations, the potential energy stored in compressor tanks is released in a controlled manner in the form of kinetic energy. Compressed air systems are used in robotics, 3D industrial printers, high-voltage pneumatic circuit breakers, conveyor systems, and a variety of other industrial machines. In the fabrication of electrical and electronic devices, the use of air compressors is very common. Cleaning printed circuit boards (PCBs) is one of the compressed air's most common applications in electronics. After production, compressed air is a non-obstructive, low-impact, and non-abrasive way to remove unwanted matter from these sensitive electronic components. In the manufacturing industry, pick-and-place machines are also used to transport items from one process to the next. These pick-and-place machines are also powered by compressed air. A tubing system delivers controlled amounts of compressed air to specific parts of the robot, which is efficiently converted into mechanical energy. In many applications, pneumatic systems are seen as a viable replacement for servo, electric, and hydraulic systems.

3.6.7 Food and Beverage

Compressed air has a wide range of uses in a variety of industries. Air compressors are used in the food and beverage industry in production chains, packaging, and

cleaning [29]. A contact system is compressed air that comes into direct contact with food products. To ensure food safety, this compressed air must be properly purified and filtered. It's also important to keep the dewpoint of contact compressed air at the right level to avoid microbial growth. The British Compressed Air Society recommends a pressure dewpoint of -40 degrees Fahrenheit for contact compressed air in their Compressed Air Best Practice Guideline.

In the beverage sector, compressed air from air compressors completes the canning of soft drink bottles and wine bottles. The compressed air will push the compression tube within the barrel to spray out the resin for coating in a brewery or winery's barrel production workshop. Other uses for an air compressor in the beverage industry include liquid transfer between barrels, grain unloading from lorry to shop, and barrel leakage testing. The air compressors also control the automatic bottle filling machine.

3.6.8 Manufacturing

An air compression system is the key source of energy that maintains a firm in manufacturing, whether it's in refineries, plastics, assembly factories, or metal fabrication [23]. Food, beverage, and pharmaceutical manufacturers use rotary screw machinery to guarantee that their goods are clean, contaminant-free, and firmly sealed. Conveyor belts, nozzles, crushers, and packing can all be powered by rotary screw machinery at the same time. Air compressors with high output help in manufacturing by:

- On the production line, using air tools
- Equipment for the welding process
- Pieces are ejected from manufacturing frames.
- Manufacturing is being monitored.
- Roller and feed equipment adjustments
- Blowing up a plastic container or a manufactured gas tank
- Conducting fundamental activities efficiently, including driving screws and rotating nuts
- Liquid padding, carton stapling, appliance sanding, dry powder conveying, and fluidizing are all done with pneumatic machines.
- Metal sandblasting and finishing

3.6.9 Aerospace

In the aerospace industry, air compressors are used in a wide range of applications [22]. Air compression that is reliable and free of contaminants is essential for the safe operation and optimal performance of aerospace equipment. Rotary screw air compressors and piston-type air compressors are the two main types of air compressors used in these applications.

- **Rotary Screw Air Compressor:** Rotary air compressors are the best choice when large volumes of air at high pressures are required regularly. Pulsation or surging of airflow is also eliminated by the compressor's continuous sweeping motion.
- **Piston Type Air Compressor:** Piston-type air compressors are best for intermittent- use applications. These units, also known as reciprocal air compressors, make excellent use of the energy they receive and tend to last longer. The reciprocating motion of the pistons, however, causes airflow pulses and surges, which may be undesirable in some applications.

3.6.10 Hospital/Medical

Many systems in medical facilities and medical air plants are powered by medical air compressors [28]. Medical air compressors are used in a variety of ways in a facility. Here are a few examples:

- **Patient Breathing:** To provide clean air to patients who are sedated or have difficulty breathing on their own.
- Compressors are used in laboratories to power equipment such as blood analyzers, mammograms, and x-rays, chiropractic tables, oxygen, and nitrogen generation, etc.
- Surgical instruments that perform puncturing, drilling, and other surgical procedures are also powered by air compressors.
- Clean air infiltration and duct systems are another application for compressors in medical facilities. The air in medical facilities must be clean and pure at all times.
- Cleaning and sterilization systems can also be powered by compressors.

3.6.11 Mining

Air compressors are widely used on every mining site [33]. They play a crucial role in making the work of miners much faster, easier, and more efficient. Mining air compressors are used to power air tools such as air picks, screwdrivers, ventilation, jackhammers, dust filtering, pneumatic hoists, driving pumps, filling cracks with cement, running punching machines, and radial percussive coal cutters. Among all air compressors, a rotary vane air compressor is highly appreciated in the mining sector. They are a reliable choice for meeting the mining industry's increasingly demanding challenges. For consistent performance across a wide range of applications, rotary vane air compressors have nearly unlimited capabilities. The rotary vane compressor is distinguished from other products by its unique design. The rotary vane principle's simplicity ensures a long life span and quiet operation. With a single offset rotor supported by two bushings, daily maintenance is simplified and unscheduled downtime is greatly reduced. A well-maintained rotary vane compressor will not degrade in performance over time. This compressor's specialized design and construction ensure minimal wear, and replacement parts are inexpensive and readily available.

3.6.12 Plastic Industries

Almost every plastic manufacturing industries are automated, which necessitates extreme precision and accuracy in process power input [29]. Air compressors are now used throughout the plastics manufacturing process, from instruments and handling of materials to manufacturing lines and machinery and facility cleaning. Rotary vane compressors are used by both large and small manufacturers in the plastics industry for a variety of applications. Rotary vane compressors also save resources and costs, which makes a lot of difference in this growing industry.

3.6.13 Power Generation

The high-pressure steam generated by the steam boiler drives the turbine, which then drives the generator to generate electricity in most factories that burn chemical fuel [29]. Boilers that use coal or fuel oil require a complete ash removal system. Because combusting the petrochemicals will produce a lot of dust. Before releasing the gas into the atmosphere, the harmful and polluting gas, primarily sulfur dioxide, must be removed. The use of an air compressor in a power plant will successfully resolve the issues mentioned above.

3.6.14 Wood Products

Without air compressors, carpentry and furniture construction would be far more difficult for everyone from beginners to seasoned carpenters. While most people are aware of pneumatic tools that are specifically intended to be powered by compressed air, such as nail guns, many people are surprised by how much equipment is specifically intended for compression. Sandblasters are used to prepare the surface by removing excess rust or dust, as well as other defects. Air sanders, like other electric or mechanical sanders, use air compression which makes these tools ideal for long-term uses. Although there is plenty of additional woodworking equipment that uses air compression.

3.6.15 Pharmaceuticals

Antibiotics are a miraculous cure for a variety of diseases. When taking antibiotics, a lot of compressed air is required during the fermentation process [29]. During the production of medicine, an air compressor is critical for supplying oxygen to microorganisms. The air compressor is also an auxiliary piece of equipment that aids in the spray dryer's transition and drying. The moisture is evaporated after the concentrated liquid is jetted into the diffuser to become the hot airflow, and the remaining solid power is put into the collector. Medicines are also packed with compressed air. Medicine in powder form is blown into a thin fibrous capsule in a pharmaceutical factory. Compressed air is also useful for filling and sealing various plastic containers.

3.6.16 Spray Painting

In both personal and professional spray painting, small air compressors are used for power airbrushes [23]. For intricate tasks such as picture editing, painting nails,

or fine art, airbrushes are utilized instead of brushes. Spraying with an air cannon necessitates the use of heavier equipment. It's often used to apply a uniform coating of liquid on huge surfaces.

3.7 Machine Learning

Machine learning employs mathematical algorithms to learn and analyze data in order to make future predictions and judgments. The term "machine learning" was coined in 1950 by Alan Turing, a pioneering computer scientist, in response to the question "Can a machine think?" [1]. In 1957, Frank Rosenblatt developed the Perceptron model, which is the first neural network [2]. The perceptron algorithm was created to identify visual inputs and divide the group into two categories. Bernard Widrow and Marcian Hoff built two neural network models in 1959, Adeline and Madeline, that could recognize binary patterns and reduce echo on phone lines, respectively [3]. The "nearest neighbor" technique was developed in 1967, allowing computers to perform very basic pattern recognition [4]. Gerald Dejong proposed the notion of explanation-based learning in 1981, in which a computer examines data and produces a general rule to exclude irrelevant information [5]. During the 1990s, machine learning research evolved from a knowledge-based to a data-driven approach. Scientists began creating programs for computers to analyze large amounts of data and learn from the results. In recent decades, machine learning has grown at an exponential rate. Our computers' ability to process and interpret data will grow in tandem with the amount of data we produce.

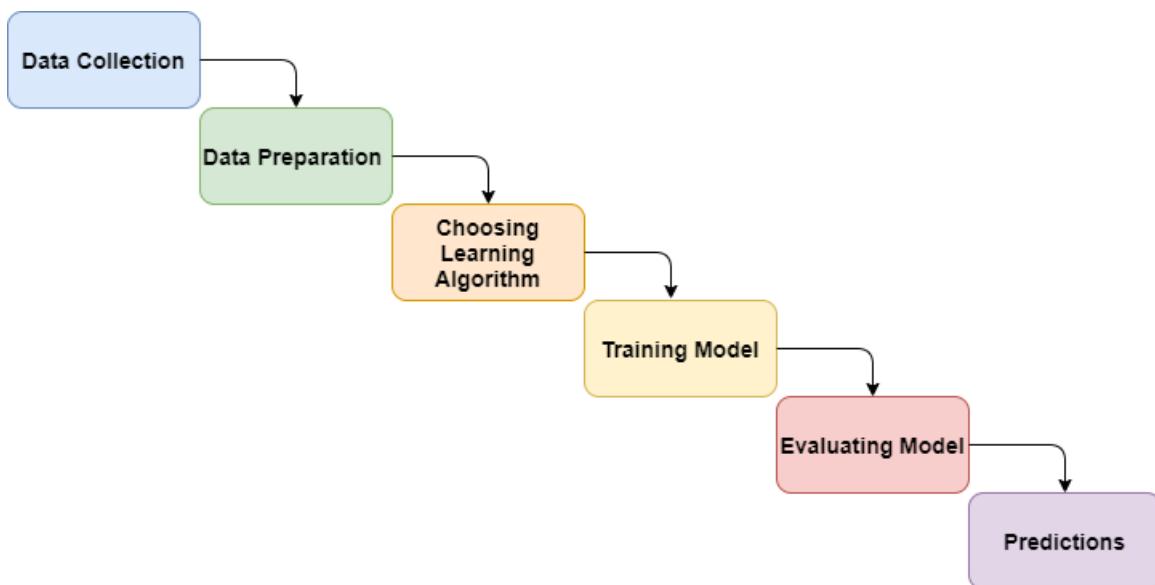


Figure 3.8: Machine Learning Workflow

3.7.1 KNN

The k-NN classification algorithm determines the k-nearest neighbor(s) and classifies numerical data records by utilizing any distance calculating methods such

as(Euclidean distance, Manhattan distance, Hamming Distance, Minkowski Distance) to calculate the distance between the test sample and all of the training samples [20].

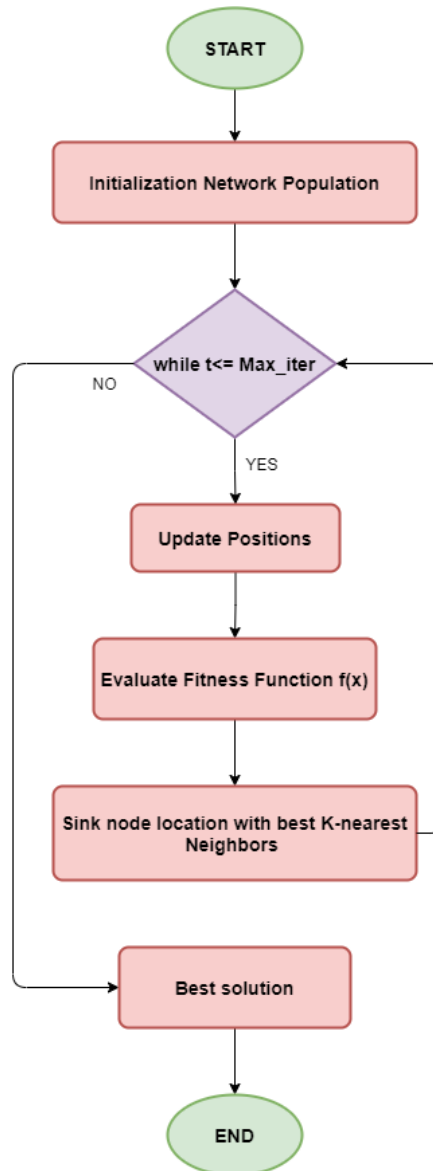


Figure 3.9: KNN Workflow

In KNN, first, it calculates the distance between the new sample and the training sample, then finds the nearest K neighbors; then, determines the category of the new sample based on the category to which the neighbor belongs. It is a supervised machine learning algorithm as the target variable is known. The k-nearest neighbor method saves all existing data and classifies fresh data points based on their similarity (e.g., distance functions) [26]. When new data appears, this is what it signifies. Then, using the K-NN method, it may be easily sorted into a suitable category.

In "Figure 3.10", there are two classes, Class A and Class B, and we have a new unknown data point "?" to find out in which of these classes will this data point belong to. The data point is identified by a majority vote of its neighbors, with the data point being assigned to the most common class among its K closest neighbors as determined by a distance function.

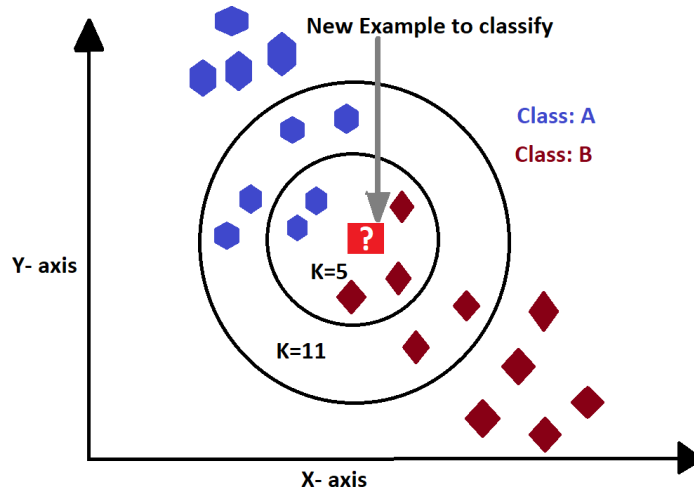


Figure 3.10: KNN Classification

Here we can see that if $k = 5$, the data point's closest three neighbors are discovered using the distance function, and the data point is categorized into class B using the majority votes of its neighbors. When $k = 11$, the data point in the preceding diagram is classed as Class A based on the majority votes of its neighbors.

There are some drawbacks to the KNN model, such as its accuracy being dependent on data quality, and the prediction step being slow with massive data. It is extremely sensitive to data scale and irrelevant attributes. It also needs a large amount of memory to store all of the training data. It can be computationally expensive because it stores all of the training data [26].

3.7.2 Random Forest

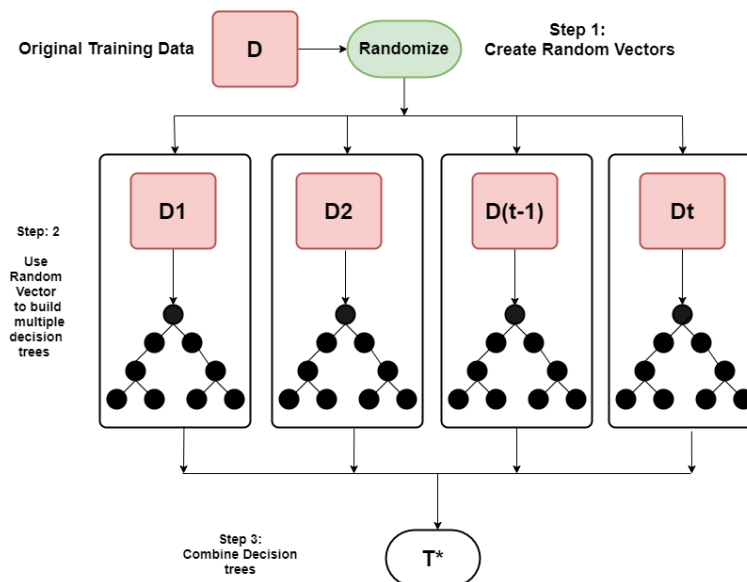


Figure 3.11: Random Forest - Flow Diagram

The random forest machine learner is made up of a large number of individual learners (trees) [32]. Multiple random tree classifications are used in the random

forest to vote on an overall classification for a given set of inputs.

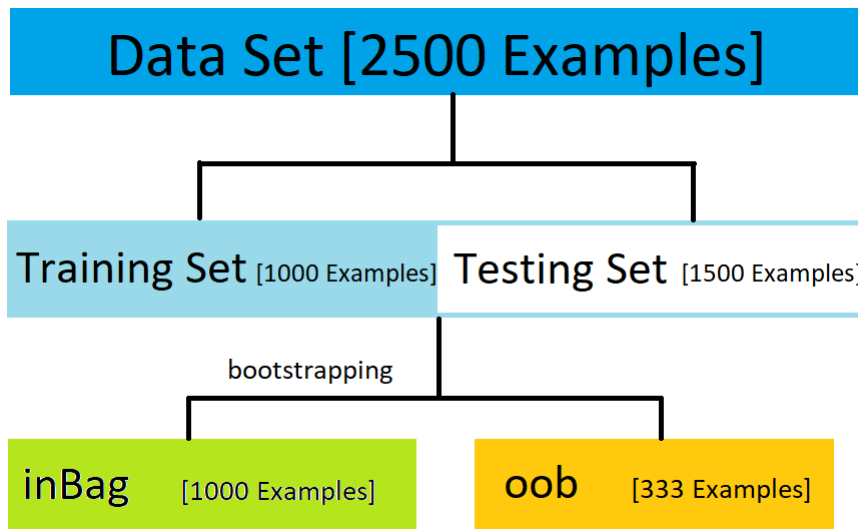


Figure 3.12: Random forest sample with replacing

The technique of "bootstrapping" is used to create a data set [inbag] by sampling replacement members from the training set. The [inbag] data set has the same number of examples as the training data set. It's possible that this fresh data collection contains duplicates from the training set. One-third of the training set data is generally missing from the [inbag] when using the bootstrapping strategy. Out-of-bag data [oob] refers to the data that has been left over. Each tree is given a random number of properties. Using typical tree-building algorithms, these properties are used to create nodes and leaves. Without pruning, each tree is allowed to reach its full potential. This technique is repeated in order to create a number of different random tree learners. The out-of-bag samples are used to test individual trees as well as the complete forest once the tree has developed. The out-of-bag error estimate is the average misclassification over all trees. This error estimate is useful for estimating machine learning performance without using the test set example. This data could be used to calculate the weights of individual tree classifications in the weighted random forest learner.

Chapter 4

Methodology

4.1 Workflow

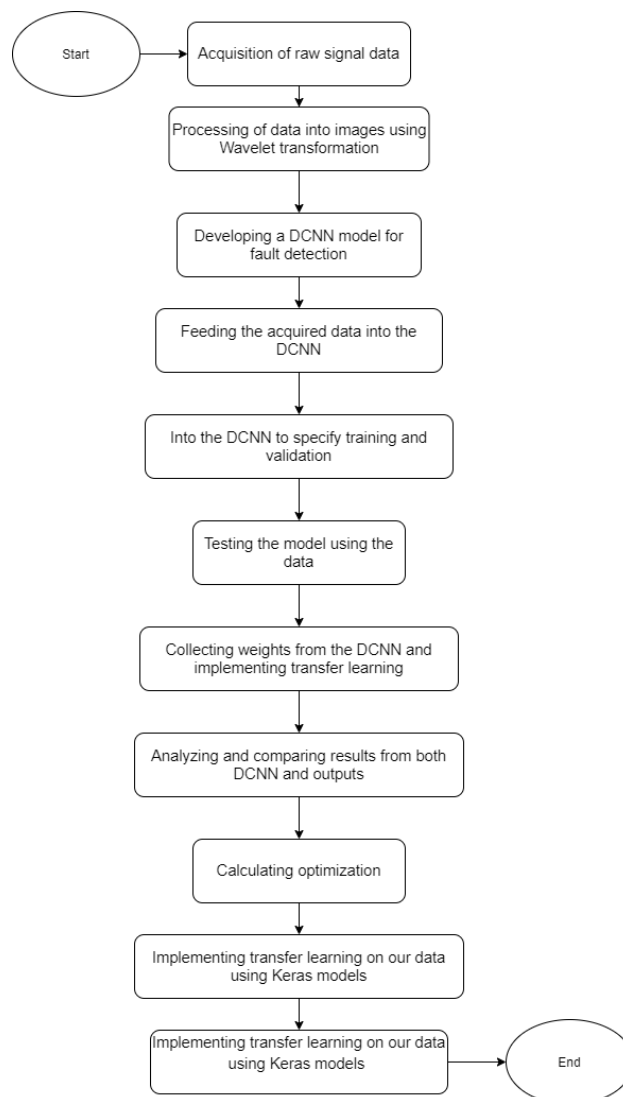


Figure 4.1: Workflow Diagram

Data has been taken from dataset repository (Verma et al., 2016). Acoustic dataset was collected on a single stage reciprocating type air compressor [15]. The air compressor's specifications are as follows:

- Air Pressure Range: 0-500 lb/m², 0-35 Kg/cm²
- Induction Motor: 5HP, 415V, 5Am, 50 Hz, 1440rpm
- Pressure Switch: Type PR-15, Range 100-213 PSI

The dataset has eight states, including a healthy state and seven faulty states, including the Leakage Inlet Valve (LIV) fault, the Leakage Outlet Valve (LOV) problem, the Non-Return Valve (NRV) problem, the Piston ring fault, the Flywheel problem, the Rider belt fault, and the Bearing problem. In every category there are 50000 acoustic data. For our uses we have triple stacked the data and keep 102400 data to convert it (32 32 100) for getting 100 images for each category. Then we process the data into images using Wavelet transformation* *

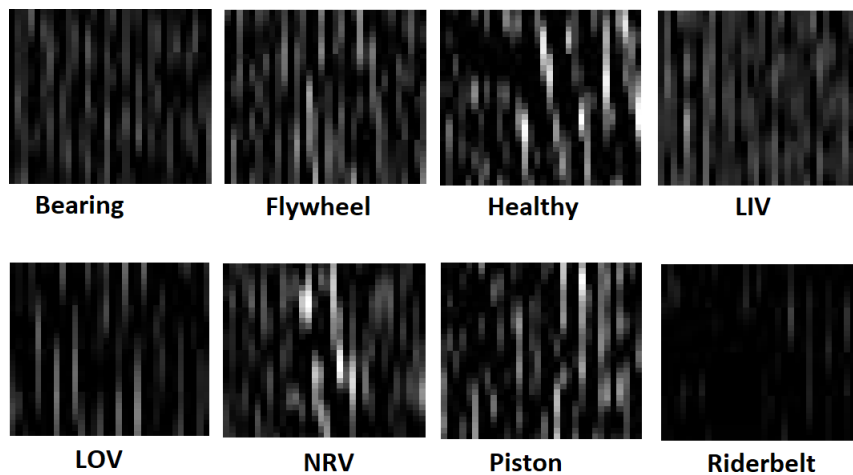


Figure 4.2: Sample Data

Normalizing the data, removing null values, and encoding class variables are all part of the data preprocessing. Then the dataset is divided into training and testing part, with 70 percent of each category for training and the remaining 30 percent being used for testing.

```
'DATA SPLITTING'
# Exp1
# Split the train and the validation set for the fitting
X_train, X_val, Y_train, Y_val = train_test_split(set3, T_L, test_size = 0.3, random_state=random_seed)
```

Figure 4.3: Data Splitting

Then we feed this data in our DCNN model. Then we specify the training and validation part of the DCNN. Then we test the model using the data. We save the model to transfer the weights for the new datasets. Then we use the save model for implementing the transfer learning. Later on we analyze and compare the results from both DCNN and transfer learning outputs. We also implement transfer learning on our data using Keras models. Finally we compare the outputs from both our model and Keras model.

4.2 Architecture Used

4.2.1 Lenet-5

In our work, we have used the Lenet-5 architecture (LeCun et al, 1998) for convolutional neural networks. Lenet-5 or Lenet architecture was first proposed in “Gradient-Based Learning Applied to Document Recognition” (1998) by French computer scientist Yann LeCun. This architecture was developed to recognize 32x32x1 greyscale images containing one digit. The author took a set of 32x32x1 greyscale images and fed them into the model. Then 6 convolutional filters of kernel size 5x5 were used to convolve the images with stride size 1. The output from this operation converted the images into another image of size 28x28 with 6 channels. As the output of the convolution operation is defined as $[(32-5)/1+1]$ and the number of channels is equal to the number of filters. After that, the average pooling layer was applied to the output from the first convolutional layer. The author used a pool size of 2 and a stride of 2. After this operation, the images were reduced to 14x14x6 images as the pooling operation shrunk the images to half their previous size. Then the next convolutional layer is applied to the images with 16 same 5x5 filters. Similarly, the output of the layer modified the data into $[(14-5)/1+1]$ or 10x10x16 images. Then the images were shrunk again using average pooling resulting in 5x5x16 images as the pooling layer was the same as before. Then the 5x5x16 data were flattened. This operation generated a vector of (5x5x16) or 400 components. These components ultimately were the neurons. These flattened neurons were then fed into a dense deep neural network. The first hidden layer of the network had 120 neurons. The neurons were fully connected. The next hidden layer comprises 84 neurons. Finally, the author used Euclidean Radial Basis Function units (RBF) to squash the neurons into the 10 specification classes. The activation functions among the hidden layers were tanh activation functions. The RBF units interpreted the results as an unnormalized negative log-likelihood of a Gaussian distribution over the last fully connected layer. The Lenet-5 architecture was dubbed as it is because it had 5 layers.

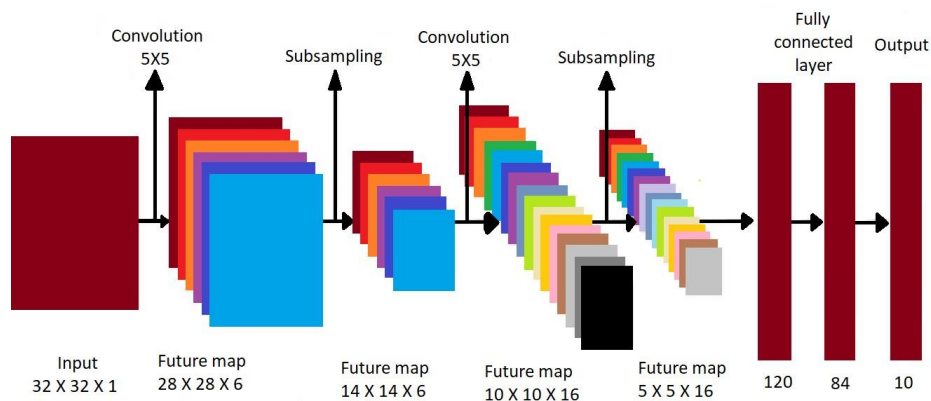


Figure 4.4: Internal Structure of Lenet-5 Architecture

4.2.2 Resnet50

Residual Neural Network or Resnet (He et al, 2015) was first proposed by a group of researchers at Microsoft. The model was generated to find a better alternative to the VGG19 (Simonyan et al, 2014) architecture. The network was based upon VGG19 and was modified accordingly. The model was first converted into a 34 layer plain image classification model and then implemented its residual properties. The architecture was used to classify the imagenet dataset. The model takes in $1 \times 224 \times 224 \times 3$ RGB color images. It then applies 64 convolution applications using filters of kernel shape 7×7 , $3 \times 3 \times 3$ padding, and 2×2 strides. For the padding, the inputs are converted into $1 \times 230 \times 230 \times 3$. After the convolution impact $[230 - (7 - 1)]$ and stride impact $224/2$, the final shape of the output becomes $1 \times 112 \times 112 \times 64$. The values within the batches are then normalized by the batch normalizer. The normalized neurons are then activated using ReLU. The neurons are then sent into the max-pooling layer. The pooling size is 3×3 , padding is $1 \times 1 \times 1$, and stride size is 2×2 . Similar to the convolution layer, after padding, the data are converted into $1 \times 114 \times 114 \times 64$. After the kernel impact $[114 - (3 - 1)]$ and stride impact $112/2$, the output is shrunk to $1 \times 56 \times 56 \times 64$. Moving on from here, the tensors are convoluted in two parallel methods and summed after the sequence. One sequence consists of two 64 filter convolutions and one 256 filter convolutions with 2 ReLU and 3 batch normalizations in between. The other sequence only has one 256 filter convolutions with one batch normalization. The batch normalization outputs are summed and turned into an output of shape $1 \times 56 \times 56 \times 256$. At some point in the network, the number of filters in the convolutions is increased to 512 and the tensor channels are also subsequently changed. The shape of the tensor becomes $1 \times 28 \times 28 \times 512$. Similarly, after some more layers, the convolution filters are increased to 1024 and the tensor is converted to $1 \times 14 \times 14 \times 1024$. The number of filters is raised to 2048 and the tensor becomes of size $1 \times 7 \times 7 \times 2048$. Finally, before generating the output probability distribution using softmax, the ReLU activated neuron of size $1 \times 7 \times 7 \times 2048$ is downsized using an average pooling layer of kernel 7×7 and stride length 1×1 . This operation generates an output of 1×2048 . Then a general matrix multiplication is applied to the tensor before feeding it to the softmax function. Finally, the softmax function gives an output of size 1×1000 .

Above figure shows the in-depth details of each layer within resnet and also denotes the modification made to VGG19 by the researchers.

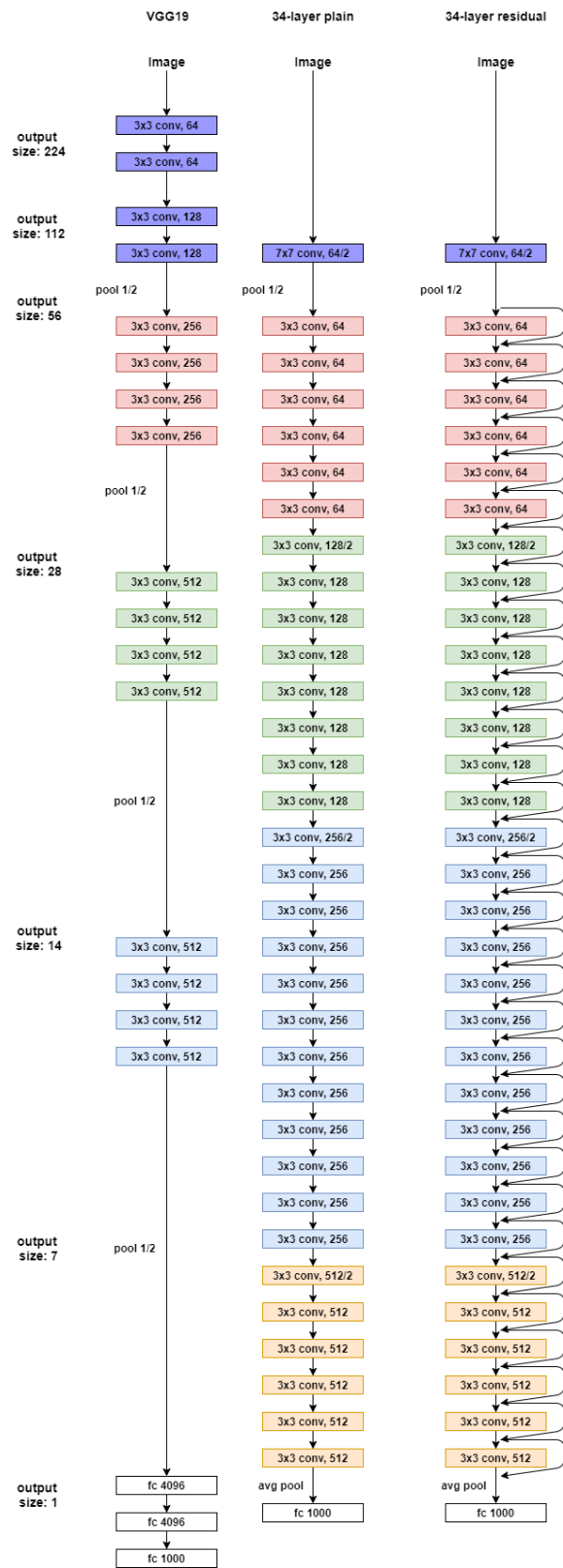


Figure 4.5: Each Layers of Resnet50 Architecture

Chapter 5

Implementation of Proposed Models

5.1 Lenet-5 Based 2D Deep CNN

We derived our model suitable for our task from the base architecture of lenet-5. Our model comprises two convolution operations, two subsampling layers, and two dense fully connected layers. The model takes greyscale images of size 32x32x1 as input. It then applies a convolution operation with 32 5x5 filters. For our model, we have decided to use padding size which is the same as the filters. Therefore, after the first convolution layer, our model gives us an output of size 32x32x32. The equation $\frac{W-F+2P}{S} + 1$ helps us find the shape of the output.

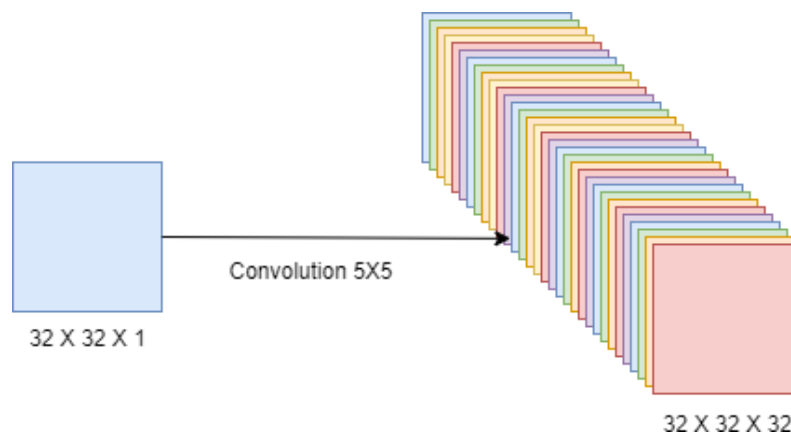


Figure 5.1: First Convolution Model

```
[ ] model = Sequential()
#Conv1
model.add(Conv2D(filters = 32, kernel_size = (5,5),padding = 'Same',
activation = 'relu', input_shape = (32,32,1)))
```

Figure 5.2: Creating First Convolution Model

After that, we have sampled the image down using max pooling. We know, max-pooling takes a pixel matrix as input and then reduces its dimensionality. Subsampling or pooling divides the input into smaller patches based on the pool size. It

starts from the top-right corner of the matrix and traverses the entire matrix based on its stride length. In the case of max pooling, it takes only the maximum value or the most prominent feature within the patch. In our case, we have used max-pooling with 2x2 patch size and 2x2 stride length. From the first convolution, the pooling layer gets the input of size 32x32x32. Following the same equation as mentioned above, max-pooling reduces this down to 16x16x32 size images.

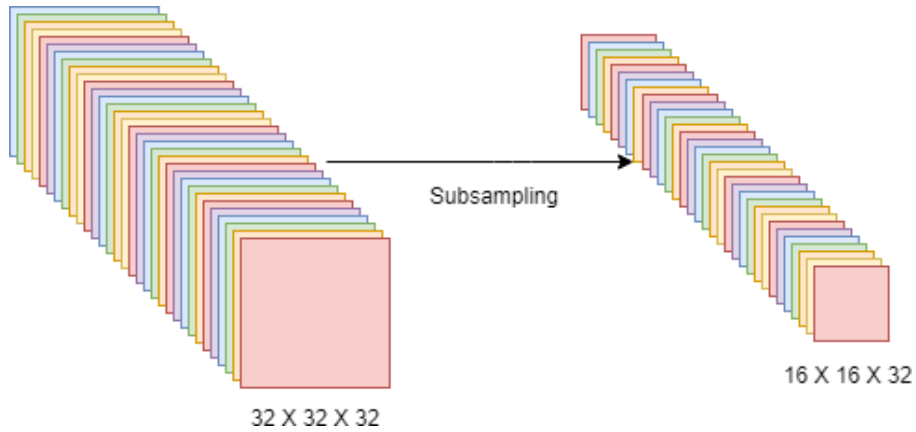


Figure 5.3: First Subsampling Layer

```
#subSampling1
model.add(MaxPool2D(pool_size=(2,2)))
model.add(Dropout(0.25))
```

Figure 5.4: Creating First Subsampling Layer

For our next convolution layer, we have decided to use 64 filters with a 3x3 kernel size and again with the same padding. After the layer is finished, the shape of our weights becomes 16x16x64.

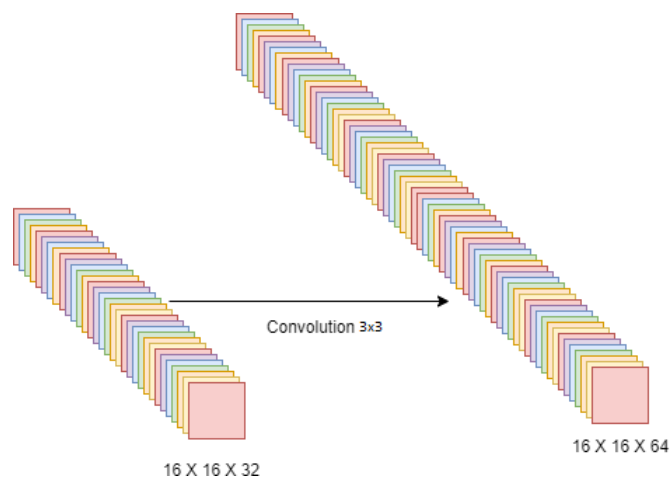


Figure 5.5: Second Convolution Model

After the second convolution layer, we have used an activation function to activate the neurons. For our purpose, we have used rely on as the activation function. ReLU

```
#Conv2
model.add(Conv2D(filters = 64, kernel_size = (3,3),padding = 'Same',
activation = 'relu'))
```

Figure 5.6: Creating Second Convolution Model

or Rectified Linear unit is an activation function that takes any value as input and returns an output after comparing it with 0. The governing function for relu can be defined as,

$$f(x) = \max(0, x) \tag{5.1}$$

Therefore, if the input is 0 or greater than 0, relu returns the input as it is. For negative inputs, relu simply outputs 0. This activation function takes the most positive values. Hence, the more positive a neuron is, the more activated it is using relu.

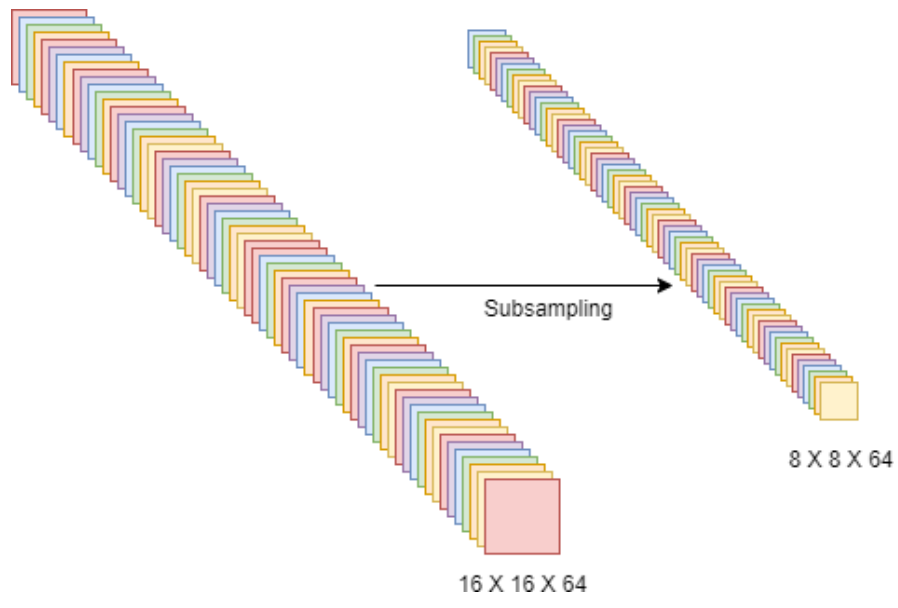


Figure 5.7: Second Subsampling Layer

```
#subSampling2
model.add(MaxPool2D(pool_size=(2,2), strides=(2,2)))
keras.layers.GlobalMaxPooling2D(data_format='channels_last')
model.add(Dropout(0.25))
```

Figure 5.8: Creating Second Subsampling Layer

As relu does not make any changes to the input shape, we have decided to use another subsampling layer using max pooling. This has allowed us to map out the more prominent features and also has helped us to reduce the computational complexity of our network. We have used max-pooling with pool size 2x2 and stride length 2x2. Again, following the same equation of $\frac{W - F + 2P}{S} + 1$, we get a downsized feature map of size 8x8x64.

This completes our two convolutional and two subsampling layers. Now, we flatten the weights into a single column weight vector. From the final shape of the matrix,

we get 8x8x64 or 4096 components of neurons. These neurons are then connected to a hidden fully connected layer. The fully connected layer consists of 128 neurons. We have activated these neurons with relu as well.

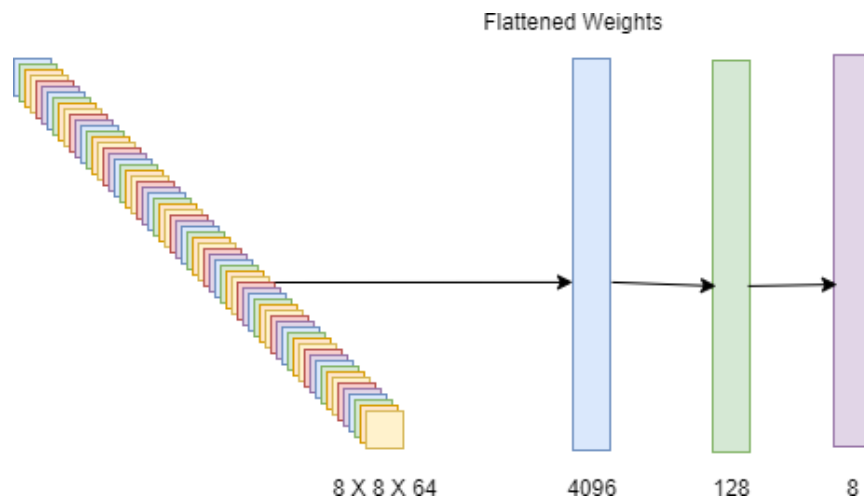


Figure 5.9: Second Subsampling Layer

```

model.add(Flatten())

model.add(Dense(128, activation = "relu"))
model.add(Dropout(0.5))

# output layer
model.add(Dense(8, activation = "softmax")) # output softmax

```

Figure 5.10: Creating Second Subsampling Layer

Finally, as shown in figure x.x, to generate our output layer with 8 classes, we have created another fully connected layer with 8 neurons. We have used the softmax activation function in this layer to activate the neurons.

Softmax or softargmax is an activation function that generates a normalized probability distribution over the predefined multi-classified output layer. Softmax operates following Luce's choice axiom (Luce, 1959). The governing equation for softmax is as follows,

$$\alpha(z)_i = \frac{s^{z_i}}{\sum_{j=1}^k s^{z_j}} \text{ for } i = 1, \dots, K \text{ and } z = (z_1, \dots, z_k) \in R \quad (5.2)$$

Here, σ is the softmax function, z is the input vector. The number of components within the vector is denoted by K and it ranges from 1 to K which is represented by i . In simple terms, softmax takes the exponents of each neuron in the input and divides it by the sum of the exponents of all the neurons. This operation normalizes the neurons and generates a probability distribution over K classes.

In our network, softmax takes each of the 8 neurons in the output layer and generates their exponents. Then it normalizes them by dividing them by the sum of their

exponents. We have then used the probabilities of each class as the output of our model.

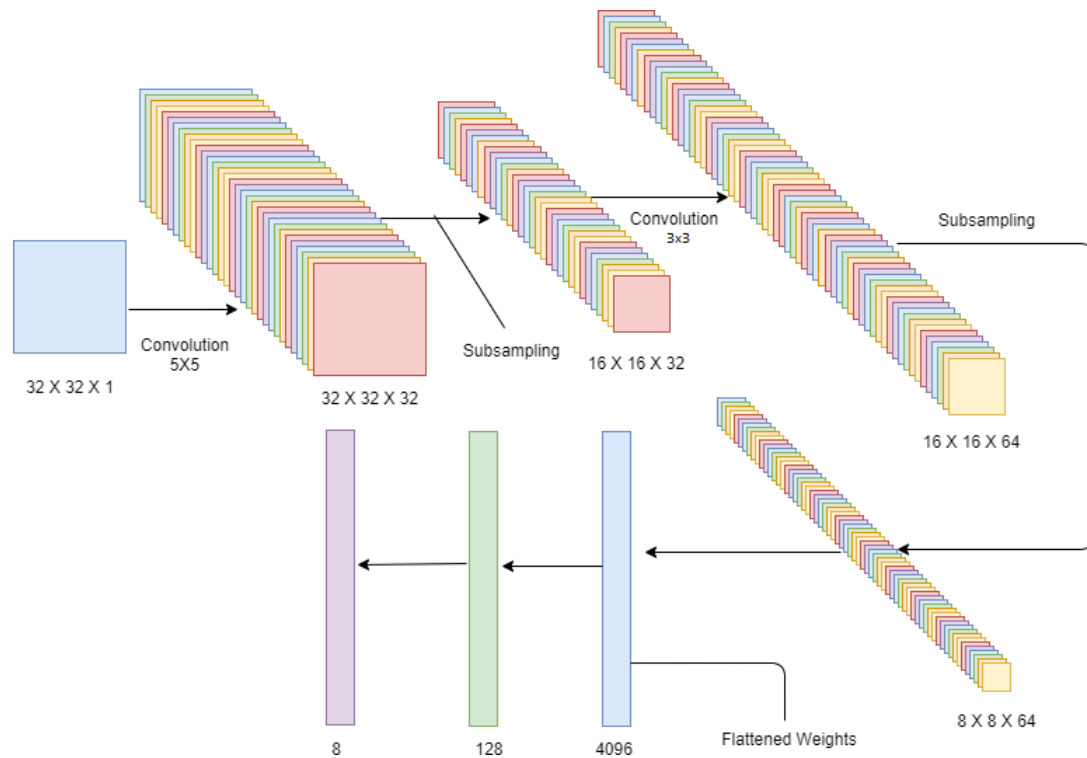


Figure 5.11: Image of Full Model

Apart from these, we have used categorical cross-entropy as our loss function and adam as our optimization function. Categorical cross-entropy has been derived from the maximum log likelihood estimation process. As we know, minimizing the classic squared error is just like maximizing the log likelihood when the error is distributed. Softmax activation function distributes the output probabilities over a Bernoulli distribution.

If we consider a model with n samples, an outcome $t_{nk} = 1$ where k is found for n th sample or 0 otherwise and a model prediction y_{nk} , we can estimate the log likelihood using the PMF of the probability distribution of the model. The PMF of the categorical distribution of the model is denoted by the given function,

$$L = \prod_{n=1}^N \prod_{k=1}^C w_k^{t_{nk}} \quad (5.3)$$

Therefore, if y_{nk} is large, $t_{nk} = 1$ is more probable or more right. Categorical cross-entropy calculates the cost simply by measuring the negative log-likelihood of the category of the outputs.

$$\text{Categorical Cross Entropy Loss} = - \sum_{n=1}^N \sum_{k=1}^C t_{nk} \log y_{nk} \quad (5.4)$$

Now, if y_{nk} is more wrong, we want loss to be larger and if y_{nk} is more right, we want loss to be smaller. For example, if we consider a case with only 1 sample, the equation for loss would be,

$$Loss = - \sum_{k=1}^K t_k \log y_k \quad (5.5)$$

If the model prediction is exactly right,

$$Loss = -\log(1) = 0$$

If the prediction has 50% probability of being on correct target,

$$Loss = -\log(0.5) = 0.693$$

If the prediction has 25% probability of being on correct target,

$$Loss = -\log(0.25) = 1.386$$

If the prediction has 0% probability of being on correct target,

$$Loss = -\log(0) = \infty$$

Following the aforementioned formula, categorical cross-entropy is measuring the loss from our model.

In order to optimize the model, we are feeding the loss function to our optimizer algorithm. For our model, we have decided to use the adam optimizer. Adam is the combination of adagrad or adaptive gradient descent and the momentum concept from stochastic gradient descent (SGD). Stochastic gradient descent can be quite effective at optimizing the loss for a single class predictor model. But for multiclass models, SGD fails to traverse the loss curve and update the weights accordingly. This is the result of having a fixed learning rate in SGD. Adagrad allows the model to have an adaptive learning rate for different parameters. The learning rate is adapted based on the sum of the square of gradients with respect to the data following the given equation,

$$\theta_{t+1,i} = \theta_{t,i} - \sqrt{\frac{\eta}{G_{t,ii} + s}} \cdot g_{t,i} \quad (5.6)$$

Here, $G_{t,ii}$ is the squared sum of the gradients.

But the problem with this adaptive learning is that the squared sum of the gradients are monotonically increased and the learning rate is slowly reaching a point where it is tending to zero and very little learning is happening. This is mitigated by the adam optimizer. Adam is the combination of adadelta and momentum. Adadelta reduces the effect of past gradients in adagrad by using a gamma weight to all the past gradients. This improves the situation exponentially and prevents the learning rate tanking to zero. It is then further improvised with the addition of momentum. Adam uses an expected value from the past gradients. The equation governing adam is as follows,

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{v_t + s}} m_t \quad (5.7)$$

This means adam starts slowly at the beginning but with more and more iterations, it picks up speed similar to the concept of momentum. This way, adam can take different sized steps for different parameters. This also leads to a much faster convergence.

5.2 Implementation of Transfer Learning on our 2D CNN

Once the model has been trained and validated on set3, we have saved our model as an .h5 file. Then we have imported our pre-trained model and have collected all the optimized and updated weights from it. After that, we have fed set2 into our model. This then classified the data within set2 based on the weights generated in the 2D CNN.

```
model=tf.keras.models.load_model('faultDetection_model_3L.h5')
model.summary()
model.get_weights()
```

Figure 5.12: Implementation of transfer flow

5.3 Implementation of Transfer Learning Using Resnet50

In order to implement this model, we have imported the resnet50 model using the keras module in python. Resnet50 is a state of the art image classification model developed by researchers at Microsoft in 2015. This model is pre-trained to classify the 1000 classes within the imagenet dataset.

```
[ ] resnet = ResNet50(input_shape=IMAGE_SIZE + [3], weights='imagenet', include_top=False)
# don't train existing weights
for layer in resnet.layers:
    layer.trainable = False
```

Figure 5.13: Importing Resnet50

```
[ ] # our layers - you can add more if you want
x = Flatten()(resnet.output)
# x = Dense(1000, activation='relu')(x)
prediction = Dense(len(folders), activation='softmax')(x)
# create a model object
model = Model(inputs=resnet.input, outputs=prediction)
```

Figure 5.14: Our Layer

For our purpose, we have imported the pre-trained model and have made the final fully connected output layer false. It thereby freed us to generate a fully connected output layer with 8 neurons for our classes. We have used the regular algorithm for resnet50 and have not changed any of its parameters. Then we have used the training set of our dataset to adapt the model to our task. The model has used all the previous neurons and trainable parameters from resnet50 along with the parameters gained from our data. Then we have tested the model with our test dataset. Thus we have transferred the weights from resnet50 to our task.

Chapter 6

Results

In order to train, validate and test our model, we have fed our model with our data 500 times. After The epochs are executed, we have generated the validation accuracy, validation loss and testing accuracy, testing loss curves with respect to every epoch. Along with that, we have generated a confusion matrix depicted by a heatmap and a precision-recall chart. Using the information from the chart, we have also generated an F1 score for our model. It has allowed us to properly evaluate our model's capability to properly detect and classify each of the 8 classes.

6.1 Model Summary Results From 2D CNN

6.1.1 Model Summary

Our 2D CNN model is summarized in the given table.

Layer(type)	Output Shape	aram
conv2d 6 (Conv2D)	None, 32, 32, 32	832
<i>maxpooling2d 6(MaxPooling2</i>	None, 16, 16, 32	0
dropout 9 (Dropout)	None, 16, 16, 32	0
conv2d 7 (Conv2D)	None, 16, 16, 64)	18496
<i>maxpooling2d 7(MaxPooling2</i>	None, 8, 8, 64	0
dropout 10 (Dropout)	None, 8, 8, 64	0
flatten 3 (Flatten)	None, 4096	0
dense 6 (Dense)	None, 128	524416
dropout 11 (Dropout)	None, 128	0
dense 7 (Dense)	None, 8	1032

Table 6.1: 2D CNN model

Total params: 544,776

Trainable params: 544,776

Non-trainable params: 0

6.1.2 Validation Accuracy Curve

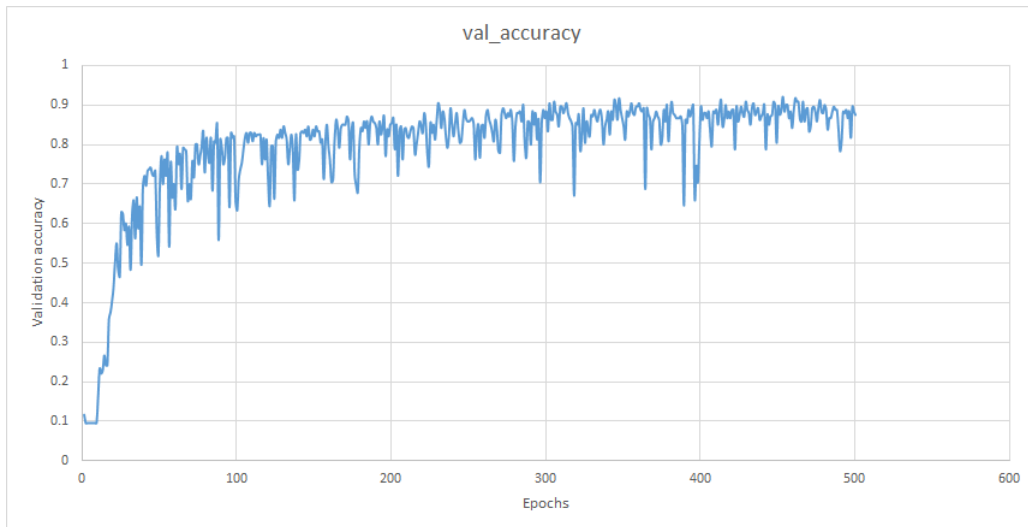


Figure 6.1: Validation Accuracy

6.1.3 Validation Loss Curve

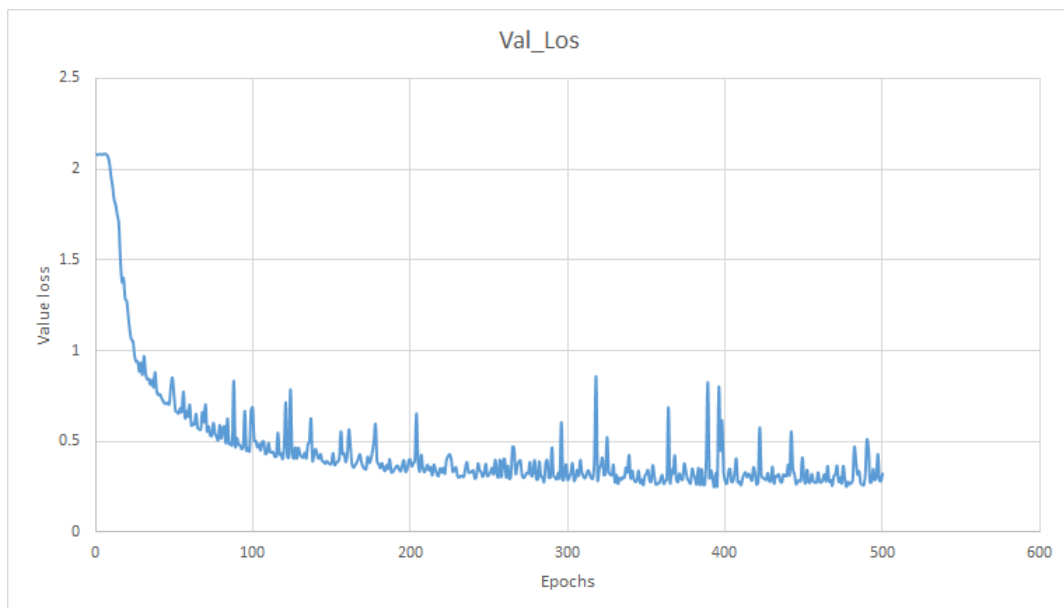


Figure 6.2: Validation Loss

6.1.4 Accuracy Curve

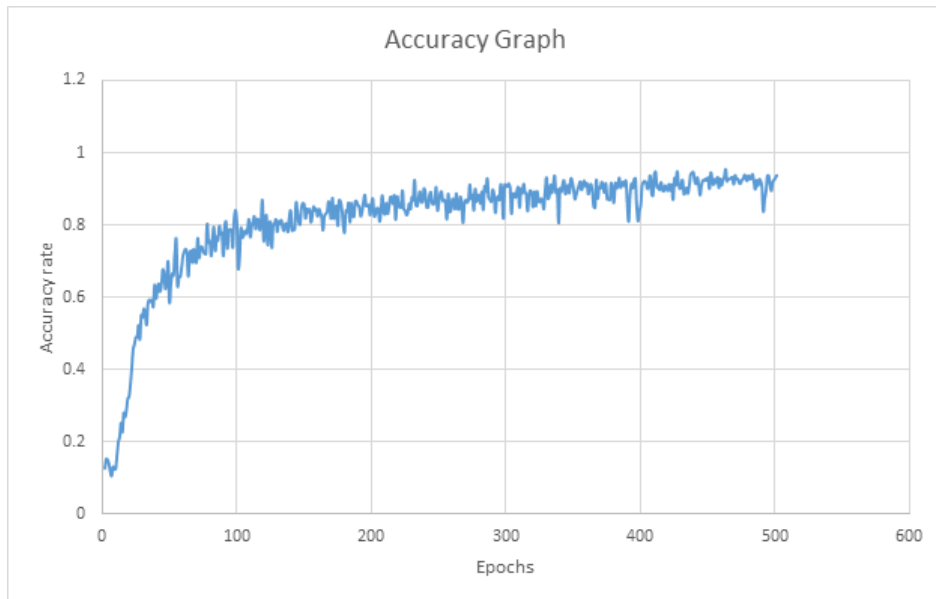


Figure 6.3: Accuracy Rate Graph

6.1.5 Testing Loss Curve

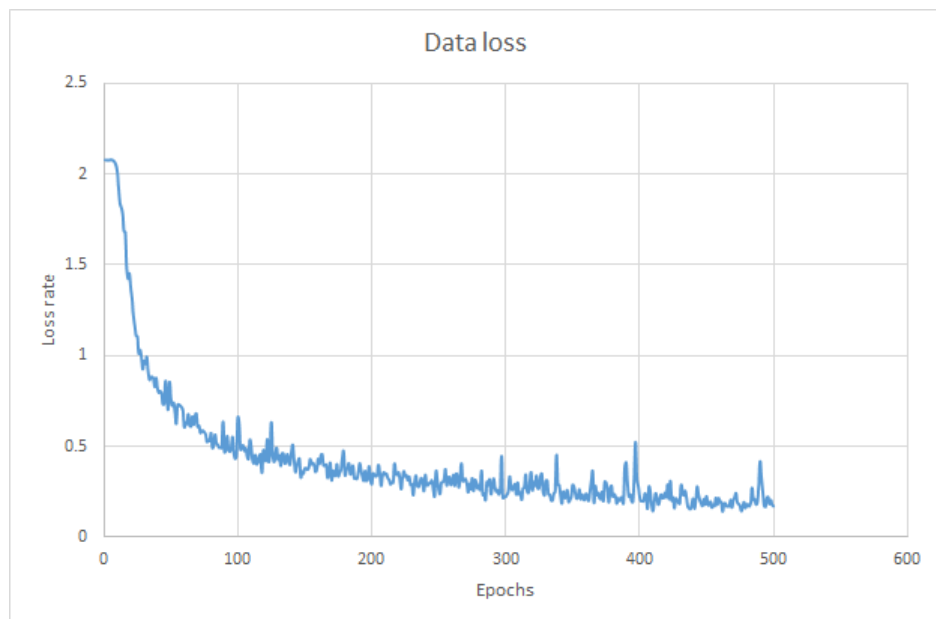


Figure 6.4: Testing Loss

6.1.6 Confusion Matrix

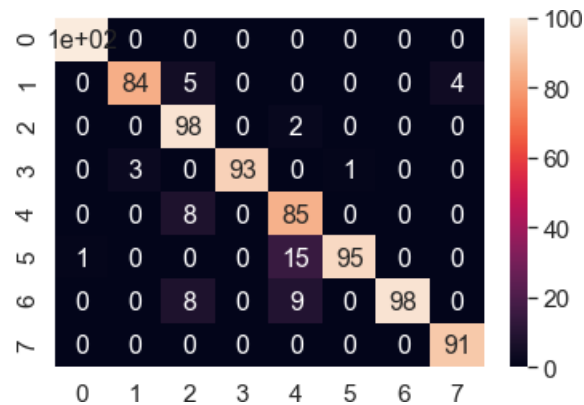


Figure 6.5: Confusion Matrix

Here, the columns represent the target classes and the rows represent the predicted classes. The values 0-7 represent the classes Bearing, Flywheel, Healthy, LIV, LOV, NRV, Piston, Riderbelt respectively.

6.1.7 Precision-Recall Calculations

Class Name	Precision	Recall	f1-score	Support
Bearing	1.00	0.99	0.99	101
Flywheel	0.84	0.97	0.90	87
Healthy	0.98	0.82	0.89	119
LIV	0.93	1.00	0.96	93
LOV	0.85	0.77	0.81	111
NRV	0.95	0.99	0.97	96
Piston	0.98	1.00	0.99	98
Rider Belt	0.91	0.96	0.93	95
Macro Avg	0.93	0.94	0.93	800

Table 6.2: Precision-Recall Table

Accuracy: 93%

6.2 Model Summary Results From Transfer Learning Framework

6.2.1 Model Summary

As we have simply forwarded the weights from our 2DCNN in our TCNN framework, the model is the same here. Therefore, the model summary is also the same.

Layer(type)	Output Shape	Param
conv2d_6 (Conv2D)	None, 32, 32, 32	832
max_pooling2d_6(MaxPooling2	None, 16, 16, 32	0
dropout_9 (Dropout)	None, 16, 16, 32	0
conv2d_7 (Conv2D)	None, 16, 16, 64)	18496
max_pooling2d_7(MaxPooling2)	None, 8, 8, 64	0
dropout_10 (Dropout)	None, 8, 8, 64	0
flatten_3 (Flatten)	None, 4096	0
dense_6 (Dense)	None, 128	524416
dropout_11 (Dropout)	None, 128	0
dense_7 (Dense)	None, 8	1032

Table 6.3: TCNN Framework

Total params: 544,776
 Trainable params: 544,776
 Non-trainable params: 0

6.2.2 Validation Accuracy Curve

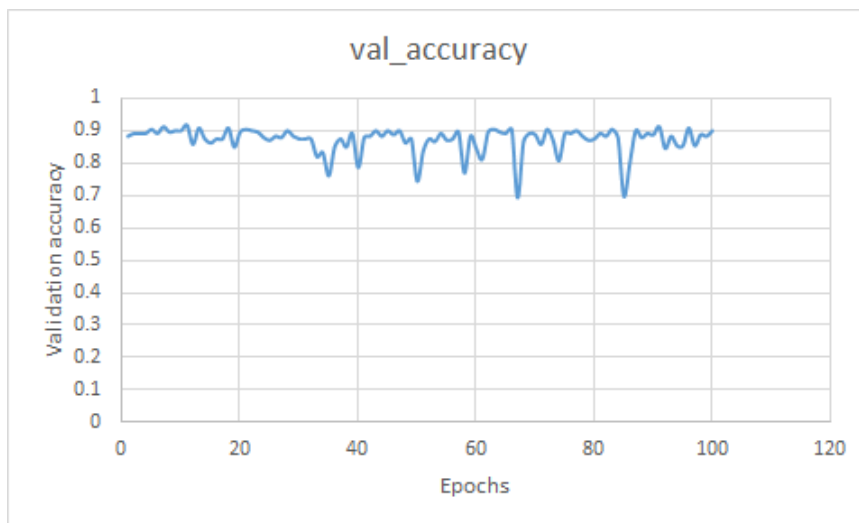


Figure 6.6: Validation Accuracy Graph

6.2.3 Validation Loss Curve

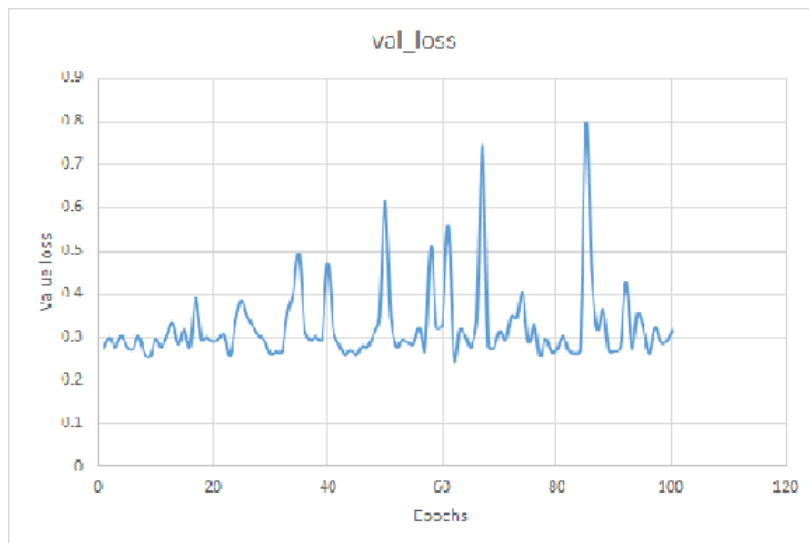


Figure 6.7: Validation Loss Graph

6.2.4 Accuracy Curve

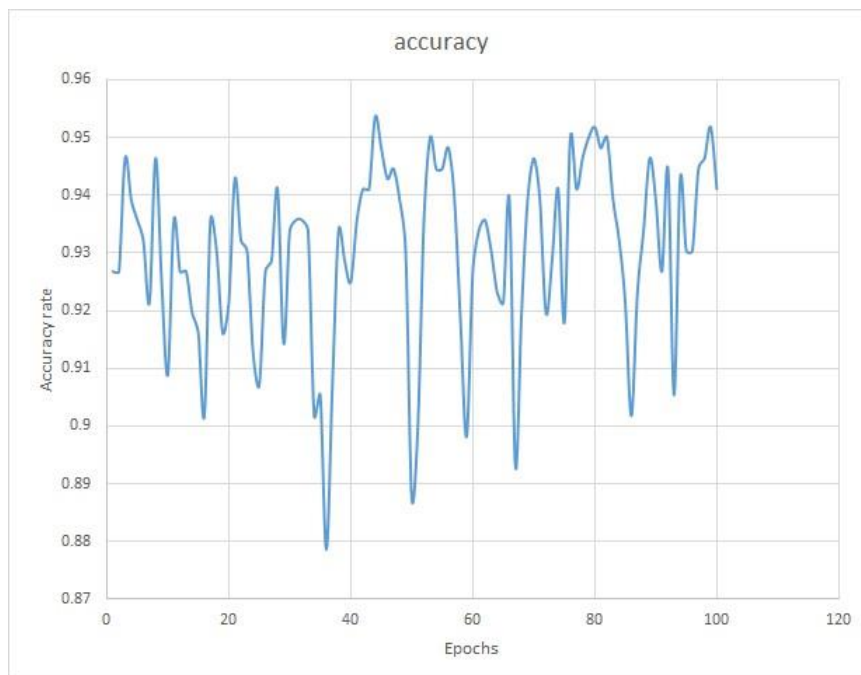


Figure 6.8: Accuracy Graph

6.2.5 Testing Loss Curve

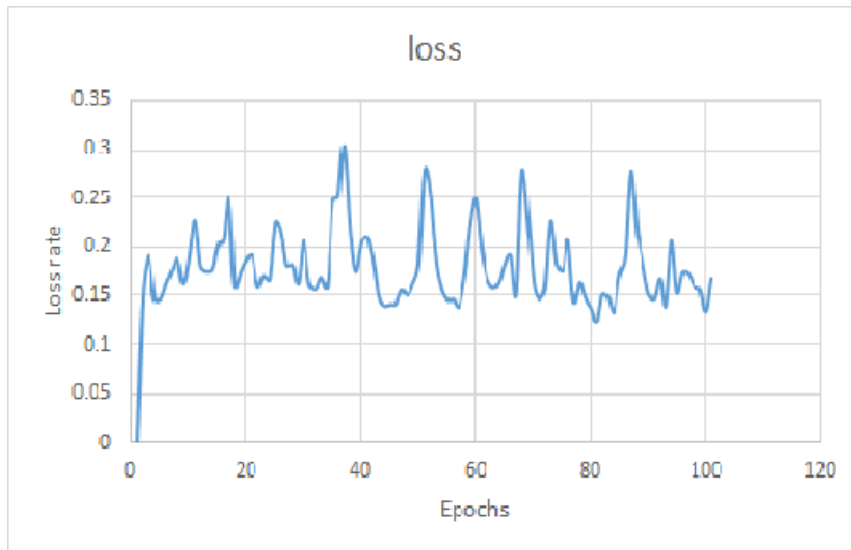


Figure 6.9: Testing Loss

6.2.6 Confusion Matrix

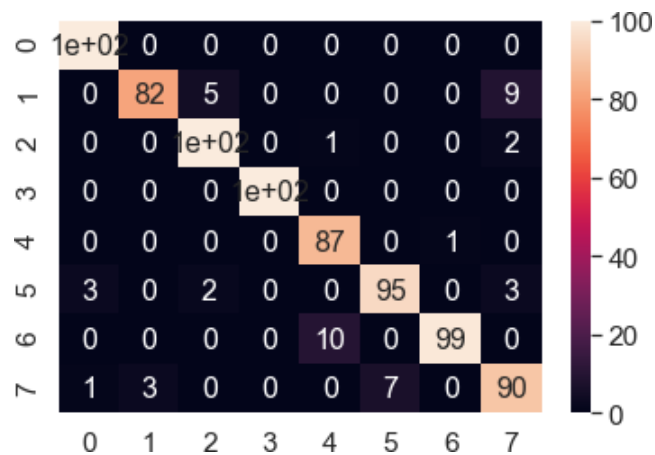


Figure 6.10: Confusion Matrix

Here, the columns represent the target classes and the rows represent the predicted classes. The values 0-7 represent the classes Bearing, Flywheel, Healthy, LIV, LOV, NRV, Piston, Riderbelt respectively.

6.2.7 Precision-Recall Calculations

Class Name	Precision	Recall	f1-score	Support
Bearing	1.00	0.96	0.98	104
Flywheel	0.82	0.96	0.88	85
Healthy	0.99	0.93	0.96	107
LIV	1.00	1.00	1.00	100
LOV	0.87	0.89	0.88	98
NRV	0.95	0.93	0.94	102
Piston	0.99	0.99	0.99	100
Rider Belt	0.90	0.87	0.88	104
Macro Avg	0.94	0.94	0.94	800

Table 6.4: Precision-Recall

Accuracy: 94%

6.3 Model Summary and Results from Transfer Learning using Resnet50

6.3.1 Model Summary

Layer (tyoe)	Output Shape	Param	Connected to
input ₁ (<i>InputLayer</i>)	None, 224, 224, 3	0	
conv1 _{pad} (<i>ZeroPadding2D</i>)	None, 230, 230, 3	0	input ₁ [0][0]
conv1 (Conv2D)	None, 112, 112, 64	9472	conv1 _{pad} [0][0]
bn _{conv1} (<i>BatchNormalization</i>)	None, 112, 112, 64	256	conv1[0][0]
activation ₁ (<i>Activation</i>)	None, 112, 112, 64	0	bn _{conv1} [0][0]
pool1 _{pad} (<i>ZeroPadding2D</i>)	None, 114, 114, 64	0	activation ₁ [0][0]
max _{pooling2d1} (<i>MaxPooling2D</i>)	None, 56, 56, 64	0	pool1 _{pad} [0][0]

...

...

Layer (tyoe)	Output Shape	Param	Connected to
add ₁₆ (<i>Add</i>)	None, 7, 7, 2048	0	bn5 _{branch2c} [0][0] activation ₄₆ [0][0]
activation ₄₉ (<i>Activation</i>)	None, 7, 7, 2048	0	add ₁₆ [0][0]
flatten ₁ (<i>Flatten</i>)	None, 100352	0	activation ₄₉ [0][0]
dense ₁ (<i>Dense</i>)	None, 8	802824	flatten ₁ [0][0]

Table 6.5: Model Summary of Resnet50

Total params: 24,390,536
Trainable params: 802,824
Non-trainable params: 23,587,712

6.3.2 Validation Accuracy Curve

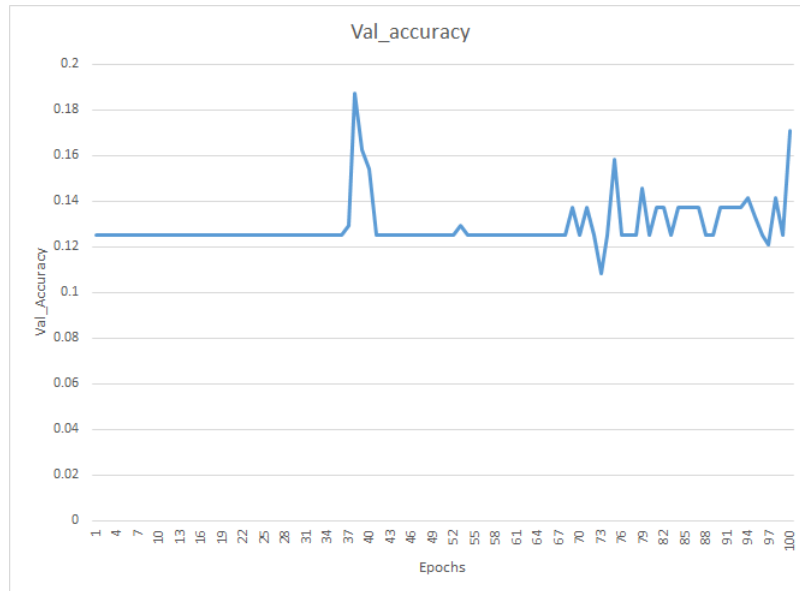


Figure 6.11: Validation Accuracy (Resnet50)

6.3.3 Validation Loss Curve

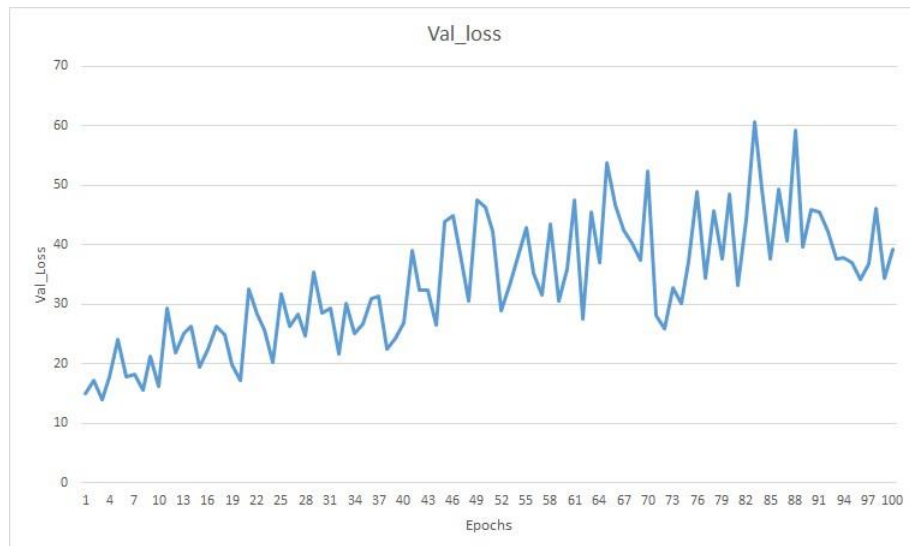


Figure 6.12: Validation Loss (Resnet50)

6.3.4 Accuracy Curve

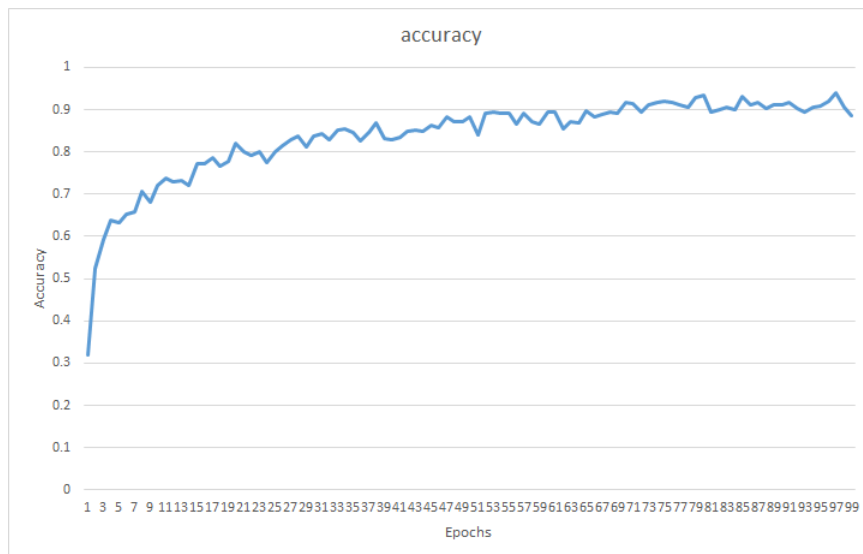


Figure 6.13: Accuracy Graph (Resnet50)

6.3.5 Testing Loss Curve

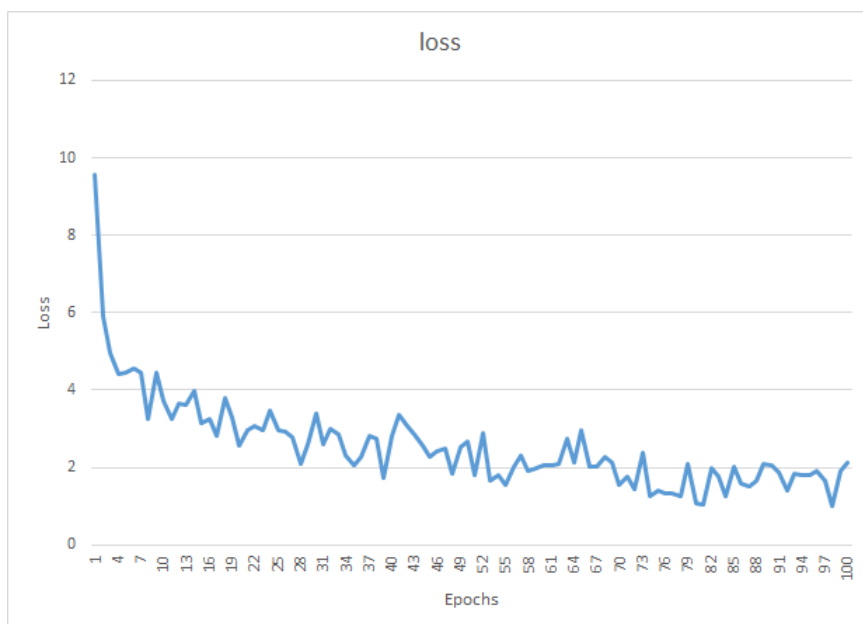


Figure 6.14: Testing Loss (Resnet50)

Chapter 7

Result Analysis

7.1 Analysis of Results from 2D CNN

If we look at the validation accuracy curve, after training our model on our training data over 500 generations, our model has successfully recognized 92% of our validation dataset correctly on average. After that, we have tested our model with our test dataset. Our testing accuracy curve shows how our testing accuracy has converged to the overall accuracy of 93% that the model has achieved.

The correlation between the predicted class and target class depicted in the confusion matrix heatmap shows the correlation among true positive (TP) and false positive (FP) outcomes of our model based on our target dataset. It paints a picture of how accurate our model has been on predicting the specified classes accordingly. The matrix also allows us to measure some more effective performance measurement metrics as well. The precision for each class denotes the rate of correct prediction for that class. Then based on the number of predictions made for that class we have calculated the recall rate for each class. Based on the precision and recall, we have generated the F1 scores as well. The F1 scores indicate the weighted average of precision and recall. Therefore, we are generating a proper performance measurement of predicting target classes taking both false positive (FP) and false negative (FN) predictions into account.

7.2 Analysis of Results from Transfer Learning Framework

If we look at the validation accuracy curve, after training our model on our training data over 100 generations, our model has successfully recognized 94% of our validation dataset correctly on average. After that, we have tested our model with our test dataset. Our testing accuracy curve shows how our testing accuracy has converged to the overall accuracy of 94% that the model has achieved.

Similar to the 2D CNN, we have generated the confusion matrix and precision-recall measurements for the transfer learning framework. If we compare the two matrices from the 2D CNN and also the transfer learning framework, we can clearly deduce the optimization acquired due to implementation of transfer learning. Moreover,

transfer learning has allowed us to achieve similar, even better results than what we have achieved from the 2D CNN. It has also reduced the computational complexity of the task by reducing the generations required by 5 times. Most importantly, the framework has allowed us to execute the task without even having to develop a complicated network from scratch, saving much valuable time and expenses.

7.3 Analysis of Results from Resnet-50 based Transfer Learning

If we look at the validation accuracy curve, after training our model on our training data over 100 generations, our model has successfully recognized 20% of the validation dataset from imagenet correctly on average. After that, we have tested our model with our test dataset. Our testing accuracy curve shows how our testing accuracy has converged to the overall accuracy of 93% that the model has achieved.

7.4 Comparison Between 2D CNN based transfer learning and Resnet-50 based Transfer Learning

Even though the overall accuracy of detecting the target classes has been quite similar between both transfer learning methods, we can clearly see the problem with resnet50 or other state of the art imagenet classifier based transfer learning frameworks when we look at the model summaries and validation losses and accuracies. We know, resnet is trained on imagenet dataset containing millions of samples of data distributed over 1000 classes. Most of these data and features extracted from those data are irrelevant for our purpose. If we look at the number of total parameters and trainable parameters in the resnet50 based architecture, it is very apparent that we are extracting and conducting many resource heavy operations on a lot of non-meaningful features. This is not at all a very efficient way of solving the task. On the other hand, if we look at our transfer learning based architecture, we can see that the model used has been tailor-made for the task. Based on the sample size of our data the total and trainable parameters in our architecture is much fewer than it is in the resnet50 based model. Along with that, the ratio of total and trainable parameters are 1:1. This is implying that no extra or useless computation is present in our architecture. Hence, our model is much more optimized and efficient in detecting air compressor faults than implementing transfer learning with other models pre-trained on imagenet.

Chapter 8

Discussion and Future Work

In our paper, we have discussed the DCCN model and how to transfer weights from a DCCN model using transfer learning. Here we can see that 2D CNN takes 500 epochs to achieve 92% accuracy to classify the data. On the other hand, transfer learning takes 100 epochs to achieve 94% accuracy to classify the data. We are able to achieve equal, if not better, outcomes using transfer learning than with the 2D CNN. It has also lowered the task's computing complexity by having fewer generations. Most importantly, the architecture allowed us to complete the task without having to create a complex network from scratch. We know that the Resnet-50 transfer learning architecture is trained on the Imagenet dataset, which has millions of data samples distributed over 1000 classes. For our purposes, the majority of these data and the attributes generated from them are unimportant.

In the outputs of our model, we can see that, there are few classes that can not be classified consistently due to fewer data. In the future, if we can get more consistent and more organized data we can work on our model for achieving greater accuracy.

Chapter 9

Conclusion

Industrial manufacturing has become the most pressing issue of the twenty-first century as a result of the world's never-ending population and demand increase. Fault detection is critical for industrial manufacturing to run safely and efficiently. We can maintain a steady manufacturing line by successfully detecting defect characteristics. As a result, developing a reliable and precise defect detection technology is now a top priority. In our paper, we have used the transfer learning model to detect industrial air compressor faults. Transfer learning improves network performance under invariant working circumstances and unifies the learning method into a single network design. Experimental results show that our proposed model is achieving an average 94% accuracy to classify the data. Furthermore, the suggested model is outperforming other fault detection methods such as machine learning and convolutional neural network.

Bibliography

- [1] A. M. TURING, "I.—COMPUTING MACHINERY AND INTELLIGENCE," *Mind*, vol. LIX, no. 236, pp. 433–460, Oct. 1950, ISSN: 0026-4423. DOI:10.1093/mind/LIX.236.433. eprint:<https://academic.oup.com/mind/article-pdf/LIX/236/433/30123314/lix-236-433.pdf>. [Online]. Available:<https://doi.org/10.1093/mind/LIX.236.433>.
- [2] F. Rosenblatt, "Perceptron simulation experiments," *Proceedings of the IRE*, vol. 48, no. 3, pp. 301–309, 1960. DOI:10.1109/JRPROC.1960.287598.
- [3] B. Widrow and M. Hoff, "Associative Storage and Retrieval of Digital Information in Networks of Adaptive "Neurons"," *Biological Prototypes and Synthetic Systems*, pp. 160–160, 1962. DOI:10.1007/978-1-4684-1716-6 { }25.
- [4] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, 1967. DOI:10.1109/TIT.1967.1053964.
- [5] G. Dejong and R. Mooney, "Explanation-based learning: An alternative view," *Machine Learning*, vol. 1, no. 2, pp. 145–176, 1986. DOI:10.1007/bf00114116.
- [6] J. Liu, Y. Liu, J. Cheng, and F. Feng, "Extraction of gear fault feature based on the envelope and time-frequency image of s transformation," *Chemical Engineering Transactions*, vol. 33, pp. 55–60, Jul. 2013. DOI:10.3303/CET1333010. [Online]. Available:<https://www.cetjournal.it/index.php/cet/article/view/CET1333010>.
- [7] U. C. Md Rifat Shahriar Tanveer Ahsan, "Fault diagnosis of induction motors utilizing local binary pattern-based texture analysis," *EURASIP Journal on Image and Video Processing*, vol. 29, no. 29 (2013), 2013, ISSN: 1687-5281. [Online]. Available:<https://rdcu.be/clwu7>.
- [8] R. Walker, S. Perinpanayagam, and I. Jennions, "Rotordynamic Faults: Recent Advances in Diagnosis and Prognosis," *International Journal of Rotating Machinery*, vol. 2013, pp. 1–12, 2013. DOI:10.1155/2013/856865.
- [9] M. Farzam Far, A. Arkkio, and J. Roivainen, "Electrical fault diagnosis for an induction motor using an electromechanical fe model," English, in *2014 International Conference on Electrical Machines (ICEM)*, United States: IEEE Institute of Electrical and Electronic Engineers, Sep. 2014. DOI:10.1109/icelmach.2014.6960440.
- [10] M. Kang and J.-M. Kim, "Reliable fault diagnosis of multiple induction motor defects using a 2-d representation of shannon wavelets," *IEEE Transactions on Magnetics*, vol. 50, no. 10, pp. 1–13, Oct. 2014, ISSN: 1941-0069. DOI:10.1109/TMAG.2014.2316474.

- [11]J. Uddin, M. Kang, D. Nguyen, and J. Kim, "Reliable Fault Classification of Induction Motors Using Texture Feature Extraction and a Multiclass Support Vector Machine," *Mathematical Problems in Engineering*, vol. 2014, pp. 1–9, 2014. doi:10.1155/2014/814593.
- [12]R. Jegadeeshwaran and V. Sugumaran, "Fault diagnosis of automobile hydraulic brake system using statistical features and support vector machines," *Mechanical Systems and Signal Processing*, vol. 52, pp. 436–446, 2015. doi: 10.1016/j.ymssp.2014.08.007.
- [13]A. Taheri-Garavand, H. Ahmadi, M. Omid, S. Mohtasebi, K. Mollazade, A. Russell Smith, and G. Carlomagno, "An intelligent approach for cooling radiator fault diagnosis based on infrared thermal image processing technique," *Applied Thermal Engineering*, vol. 87, pp. 434–443, 2015. doi:10.1016/j.applthermaleng.2015.05.038.
- [14]S. Khan and J. Kim, "Rotational speed invariant fault diagnosis in bearings using vibration signal imaging and local binary patterns," *The Journal of the Acoustical Society of America*, vol. 139, no. 4, EL100–EL104, Apr. 2016. doi: 10.1121/1.4945818.
- [15]N. Verma, R. Sevakula, S. Dixit, and A. Salour, "Intelligent Condition Based Monitoring Using Acoustic Signals for Air Compressors," *IEEE Transactions on Reliability*, vol. 65, no. 1, pp. 291–309, 2016. doi:10.1109/tr.2015.2459684.
- [16]Y. Du, Y. Chen, G. Meng, J. Ding, and Y. Xiao, "Fault severity monitoring of rolling bearings based on texture feature extraction of sparse time–frequency images," *Applied Sciences*, vol. 8, no. 9, 2018, ISSN: 2076-3417. doi:10.3390/app8091538. [Online]. Available: <https://www.mdpi.com/2076-3417/8/9/1538>.
- [17]M. Hasan and J. Kim, "Bearing Fault Diagnosis under Variable Rotational Speeds Using Stockwell Transform-Based Vibration Imaging and Transfer Learning," *Applied Sciences*, vol. 8, no. 12, p. 2357, 2018. doi:10.3390/app8122357.
- [18]M. J. Hasan, M. M. Islam, and J.-M. Kim, "Acoustic spectral imaging and transfer learning for reliable bearing fault diagnosis under variable speed conditions," *Measurement*, vol. 138, pp. 620–631, 2019, ISSN: 0263-2241. doi: <https://doi.org/10.1016/j.measurement.2019.02.075>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0263224119301939>.
- [19]P. Mohan, "An analysis of air compressor fault diagnosis using machine learning technique," *JOURNAL OF MECHANICS OF CONTINUA AND MATHEMATICAL SCIENCES*, vol. 14, Dec. 2019. doi:10.26782/jmcmms.2019.12.00002.
- [20]L. Wang, "Research and implementation of machine learning classifier based on knn," *IOP Conference Series: Materials Science and Engineering*, vol. 677, p. 052038, Dec. 2019. doi:10.1088/1757-899X/677/5/052038.

- [21]Y. Wang, B. Zhou, M. Cheng, H. Fu, D. Yu, and W. Wu, "A fault diagnosis scheme for rotating machinery using recurrence plot and scale invariant feature transform," in *Proceedings of the 3rd International Conference on Mechatronics Engineering and Information Technology (ICMEIT 2019)*, Atlantis Press, 2019, pp. 675–681, ISBN: 978-94-6252-708-9. doi:<https://doi.org/10.2991/icmeit-19.2019.108>. [Online]. Available:<https://doi.org/10.2991/icmeit-19.2019.108>.
- [22]Compressed Air Systems", *Air Compressors for Aerospace*, Oct. 2020. [Online]. Available:<https://www.compressedairsystems.com/blog/air-compressors-aerospace/>.
- [23]iSeekplant", *Top 10 Uses of Air Compressors*, Jun. 2020. [Online]. Available: <https://blog.iseekplant.com.au/blog/top-ten-uses-air-compressors>.
- [24]P. Scientific, *Nitrogen Compressor Technology - How does it work?* Oct.2020. [Online]. Available:<https://www.peakscientific.com/discover/news/nitrogen-gas-generator-compressor-technology-how-does-it-work/>.
- [25]BigRentz, Inc.", *How Air Compressors Work: An Animated Guide — Bi- gRentz*, Apr. 2021. [Online]. Available:<https://www.bigrentz.com/blog/how-air-compressors-work>.
- [26]M. Chatterjee, *A Quick Introduction to KNN Algorithm*, Apr. 2021. [Online]. Available:<https://www.mygreatlearning.com/blog/knn-algorithm-introduction/>.
- [27]G., *Single Stage Compressor — Working, Components and Applications:* Mar. 2021. [Online]. Available:<https://mechanicalboost.com/single-stage-compressor-working-components-and-applications/>.
- [28] *Medical Air Systems by DD Compressor Inc in San Jose*, Apr. 2021. [Online]. Available:<https://www.danddcompressor.com/blog/2019/09/what-is-compressed-air-used-for-in-hospitals/>.
- [29] quincycompressor", *Compressed Air in Glass Manufacturing*, Feb. 2021. [Online]. Available:<https://www.quincycompressor.com/compressed-air-glass-manufacturing/>.
- [30] *Rotary Air Compressors For The Automotive Industry — Kaishan USA*, May 2021. [Online]. Available:<https://kaishanusa.com/blog/rotary-air-compressors-for-the-automotive-industry/>.
- [31]A. Albert F., *Air compressor for charging an internal combustion engine - ARAUJO ALBERT F.* [Online]. Available:<https://www.freepatentsonline.com/6434940.html>.
- [32]F. Livingston, *Ece591q machine learning journal paper. fall 2005. implementation of breiman's random forest machine learning algorithm.*
- [33]Mattei Compressors, Inc.", *Air Compressors for the Mining Industry — Mattei.* [Online]. Available:<http://www.matteicomp.com/mining-industry-air-compressors#:~:text=In%20surface%20and%20subsurface%20mines,material%20conveyors%2C%20and%20ventilation%20systems.&text=The%20unique%20design%20of%20the,it%20apart%20from%20other%20products..>

[34]H. Yuan, F. Wen, H. W. Qu, and H. Wang, "Fault diagnosis of rolling bearings based on surf algorithm,"