People's Democratic Republic of Algeria Ministry of Higher Education

May 8th, 1945 University of Guelma-

Faculty of Science and Technology

Department of Electronics and Telecommunications Master's Thesis



**Field: Science and Technology**

**Program: Electronic**

**Specialty: Instrumentation**

Diagnosis and Detection of Faults in Rolling Machines

Using deep learning

**Presented by: Supervisor:**

**Youcef Mohammed Elfateh *BOUDJEHEM Djalil***

**June 2024**

**Dedicate**

To my parents, Whose unwavering love, support, and sacrifices made this journey possible. Your belief in me has been my driving force.

To my mentor, professor BOUDJAHEM DJALIL, Your guidance, wisdom, and patience have been invaluable throughout this process. I am forever grateful for your mentorship.

To my friends and family, Thank you for your understanding, encouragement, and for being my constant source of strength. Your presence made the challenges bearable.

**Acknowledgements**

*First of all, I would like to thank Almighty God for giving me the faith and courage to complete this project.*

*I also express my sincere gratitude to my professor, Mr.* *BOUDJEHEM Djalil, as well as the entire teaching faculty, for their valuable guidance and wise counsel throughout my university studies.*

*Their encouragement and dedication allowed me to acquire the essential knowledge and develop the necessary skills for the completion of this work.*

*I am truly grateful to them for their patience, availability, and efforts made to shape us into competent and responsible professionals.*

Table of Contents

[Abstract 1](#_Toc169727717)

[General Introduction 1](#_Toc169727718)

[I. Chapter 1 : Faults of rotating machines 3](#_Toc169727719)

[I.1. Introduction: 3](#_Toc169727720)

[I.2. components of rotating machine: 3](#_Toc169727721)

[I.2.1. Stator: 3](#_Toc169727722)

[I.2.2. Rotor: 4](#_Toc169727723)

[I.2.3. Shaft: 4](#_Toc169727724)

[I.2.4. Bearings: 5](#_Toc169727725)

[I.3. Faults Study of electrical machines 5](#_Toc169727726)

[I.3.1. Causes of faults: 6](#_Toc169727727)

[I.3.2. Importance of fault study: 6](#_Toc169727728)

[I.3.3. Classification of faults in electrical machines: 7](#_Toc169727729)

[I.3.3.1. Stator defects: 7](#_Toc169727730)

[I.3.3.2. Rotor defects: 8](#_Toc169727731)

[I.4. Fault due to the engine: 9](#_Toc169727732)

[I.4.1. Mechanical Failures: 9](#_Toc169727733)

[I.4.1.1. Bearing Defects 9](#_Toc169727734)

[I.4.1.2. Eccentricity Defects 9](#_Toc169727735)

[I.4.2. Electrical Failures: 9](#_Toc169727736)

[I.4.2.1. Stator Faults: 9](#_Toc169727737)

[I.4.2.2. Rotor Faults: 10](#_Toc169727738)

[I.4.2.3. Bar Breakage Fault: 10](#_Toc169727739)

[I.4.2.4. Breakage of a Portion of the Short-Circuit Ring: 10](#_Toc169727740)

[I.5. Conclusion 11](#_Toc169727741)

[II. Chapter 2: Deep learning 13](#_Toc169727742)

[II.1. Deep Learning 13](#_Toc169727745)

[II.2. Deep Learning function 13](#_Toc169727746)

[II.3. The prevalence of deep learning 15](#_Toc169727747)

[II.3.1. No Feature Extraction 15](#_Toc169727748)

[II.3.2. Big Data improves deep learning accuracy 16](#_Toc169727749)

[II.4. Neural Networks in Machine Learning 17](#_Toc169727750)

[II.5. Convolutional Neural Network (CNN) in deep Learning 17](#_Toc169727751)

[II.5.1. Convolutional Neural Network Design 18](#_Toc169727752)

[II.5.2. Convolutional Neural Network Training 19](#_Toc169727753)

[II.5.3. CNN Evaluation 19](#_Toc169727754)

[II.5.4. Applications of CNN 20](#_Toc169727755)

[II.5.5. Advantages of CNN 20](#_Toc169727756)

[II.5.6. Disadvantages of CNN 20](#_Toc169727757)

[II.6. Conclusion: 21](#_Toc169727758)

[III. Chapter 3: Case study and Simulation Results 23](#_Toc169727759)

[III.1. Introduction 23](#_Toc169727761)

[III.2. Fault detection data base 23](#_Toc169727762)

[III.3. The Structure of work: 26](#_Toc169727763)

[III.4. Optimization Algorithms: 30](#_Toc169727764)

[III.4.1. Stochastic Gradient Descent (SGD): 30](#_Toc169727765)

[III.4.2. Adam (Adaptive Moment Estimation): 30](#_Toc169727766)

[III.5. Experiment and results 31](#_Toc169727767)

[III.5.1. Case01 (SGD): 32](#_Toc169727768)

[III.5.2. Case 02 Adam: 34](#_Toc169727769)

[III.6. SGD and Adam Results Comparaison: 36](#_Toc169727770)

[III.6.1. Case 01 :(32×32 pixel) input data size 36](#_Toc169727771)

[III.6.2. Case 02 :(64\*64 pixels) input data size 39](#_Toc169727772)

[III.7. Conclusion: 42](#_Toc169727773)

**List of figures**

[Figure ‎I‑1:Vue of rolling machine 3](file:///E:\master%20thesis%201.docx#_Toc169727611)

[Figure ‎I‑2: stator of rolling machine 4](file:///E:\master%20thesis%201.docx#_Toc169727612)

[Figure ‎I‑3: rotor of rolling machine 4](file:///E:\master%20thesis%201.docx#_Toc169727613)

[Figure ‎I‑4: rolling Element bearings 5](#_Toc169727614)

[Figure ‎I‑5 : Classification of defects according to their origins 7](file:///E:\master%20thesis%201.docx#_Toc169727615)

[Figure ‎II‑1:AI vs. machine learning vs. deep learning 13](file:///E:\master%20thesis%201.docx#_Toc169727616)

[Figure ‎II‑2:A typical neural network 13](#_Toc169727617)

[Figure ‎II‑3:deference between deep learning and ML algorithms 15](#_Toc169727618)

[Figure ‎II‑4:Deep learning algorithms improve with increasing amounts of data. 16](#_Toc169727619)

[Figure ‎II‑5design convolution neural network 18](file:///E:\master%20thesis%201.docx#_Toc169727620)

[Figure ‎III‑1:Test stand for collecting vibration signals from REBs used by CWRU. 23](file:///E:\master%20thesis%201.docx#_Toc169727621)

[Figure ‎III‑2: Photographs of three kinds of damaged bearings 24](file:///E:\master%20thesis%201.docx#_Toc169727622)

[Figure ‎III‑3: Diagram of the data augmentation. 24](#_Toc169727623)

[Figure ‎III‑4:Time domain waveforms of bearing vibration signals (7mm) in 4 different speeds 25](#_Toc169727624)

[Figure ‎III‑5:Time domain waveforms of bearing vibration signals (21mm) in 4 different speeds 25](#_Toc169727625)

[Figure ‎III‑6:signal-to-image conversion 64×64 pixel 27](file:///E:\master%20thesis%201.docx#_Toc169727626)

[Figure ‎III‑7signal-to-image conversion 32×32 pixel 27](file:///E:\master%20thesis%201.docx#_Toc169727627)

[Figure ‎III‑8:Diagram of the 2d-DCNN model the training (SGD 32×32) 32](#_Toc169727628)

[Figure ‎III‑9:Diagram of the 2d-DCNN model the training (SGD 64×64) 33](#_Toc169727629)

[Figure ‎III‑10:Diagram of the 2d-DCNN model the training (Adam 32**×**32) 34](#_Toc169727630)

[Figure ‎III‑11:Diagram of the 2d-DCNN model the training (Adam 64×64) 35](#_Toc169727631)

[Figure ‎III‑12: confusion matrix of SGD optimizer (32×32) 36](#_Toc169727632)

[Figure ‎III‑13: confusion matrix of Adam optimizer (32×32) 36](#_Toc169727633)

[Figure ‎III‑14 confusion matrix of SGD optimizer (64×64) 39](#_Toc169727634)

[Figure ‎III‑15: confusion matrix of Adam (64×64) optimizer 39](#_Toc169727635)

**List of tables**

[**Table 1**:Description of all experimental datasets of 32×32 data size. 28](#_Toc169727601)

[**Table 2**:details about outer rigs dataset 32×32 28](#_Toc169727602)

[**Table 3**:Description of all experimental datasets of 64×64 data size 29](#_Toc169727603)

[**Table 4**:details about outer ring dataset 64×64 29](#_Toc169727604)

[**Table 5**:structure parameters and performance indicator 31](#_Toc169727605)

[**Table 6**:summarizing the classification result of SGD 32 37](#_Toc169727606)

[**Table 7**:summarizing the classification result of Adam 32 37](#_Toc169727607)

[**Table 8**:summarizing the classification result of SGD (64×64 pixel) 40](#_Toc169727608)

[**Table 9**:summarizing the classification result of Adam 64 40](#_Toc169727609)

# Abstract

This work investigates the application of deep learning techniques, specifically 2D Convolutional Neural Networks (CNNs), for the diagnosis and detection of faults in electrical rotating machines, with a focus on bearing fault detection. The study leverages a comprehensive database of vibration signals acquired from a test rig at Case Western Reserve University, comprising various bearing fault conditions and operating conditions.

The vibration signals are preprocessed and converted into image representations suitable for input to the CNN models. Two different input image sizes, 32x32 and 64x64 pixels, are explored to evaluate the impact of spatial resolution on the model's performance. Additionally, the study compares the efficacy of two widely-used optimization algorithms, Stochastic Gradient Descent (SGD) and Adam, in training the CNN models.

The proposed 2D CNN architectures are designed to automatically learn discriminative features from the input vibration signal images, enabling accurate classification of different bearing fault types, including inner race faults, outer race faults, and ball faults, as well as normal bearing conditions. Data augmentation techniques are employed to mitigate overfitting and enhance the model's generalization capabilities.

Extensive experiments are conducted, and the results of using the CNN models, achieve classification accuracies of up to 99.58% for the 32x32 input size and 98.41% for the 64x64 input size when trained with the Adam optimizer. Detailed analysis of the confusion matrices and classification metrics provides insights into the strengths and weaknesses of each configuration.

This work highlights the potential of deep learning-based approaches, particularly 2D CNNs, for bearing fault diagnosis and remote detection of faults in electrical machines. The study contributes to the advancement of predictive maintenance strategies in industrial applications and paves the way for further research in this domain.

ملخص

يبحث هذا العمل في تطبيق تقنيات التعلم العميق، وتحديداً الشبكات العصبية الالتفافية ثنائية الأبعاد (CNNs)، لتشخيص واكتشاف الأعطال في الآلات الكهربائية الدوارة، مع التركيز على اكتشاف أعطال المحامل. تستفيد الدراسة من قاعدة بيانات شاملة لإشارات الاهتزاز تم الحصول عليها من جهاز اختبار في جامعة Case Western Reserve، والتي تشمل مجموعة متنوعة من حالات أعطال المحامل وظروف التشغيل. تتم معالجة إشارات الاهتزاز مسبقاً وتحويلها إلى تمثيلات صورية مناسبة لإدخالها إلى نماذج الشبكات العصبية الالتفافية. يتم استكشاف حجمين مختلفين للصور المدخلة، 32×32 و64×64 بكسل، لتقييم تأثير الدقة المكانية على أداء النموذج. بالإضافة إلى ذلك، تقارن الدراسة فعالية خوارزميتين للتحسين واسعتي الاستخدام، وهما الانحدار التدريجي العشوائي (SGD) وآدم (Adam)، في تدريب نماذج الشبكات العصبية الالتفافية. تم تصميم هياكل الشبكات العصبية الالتفافية ثنائية الأبعاد المقترحة لتعلم الميزات التمييزية تلقائياً من صور إشارات الاهتزاز المدخلة، مما يتيح التصنيف الدقيق لأنواع مختلفة من أعطال المحامل، بما في ذلك أعطال الحلقة الداخلية، وأعطال الحلقة الخارجية، وأعطال الكرات، بالإضافة إلى حالات المحامل الطبيعية. يتم استخدام تقنيات تضخيم البيانات للتخفيف من مشكلة الإفراط في التخصيص وتحسين قدرات التعميم للنموذج. يتم إجراء تجارب مكثفة، وتحقق نتائج استخدام نماذج الشبكات العصبية الالتفافية دقة تصنيف تصل إلى 99.58% لحجم الإدخال 32×32 و98.41% لحجم الإدخال 64×64 عند التدريب باستخدام محسن آدم. يوفر التحليل التفصيلي لمصفوفات الالتباس ومقاييس التصنيف رؤى حول نقاط القوة والضعف لكل تكوين. يسلط هذا العمل الضوء على إمكانات النهج القائمة على التعلم العميق، وخاصة الشبكات العصبية الالتفافية ثنائية الأبعاد، لتشخيص أعطال المحامل والكشف عن الأعطال عن بُعد في الآلات الكهربائية. تساهم الدراسة في تطوير استراتيجيات الصيانة التنبؤية في التطبيقات الصناعية وتمهد الطريق لمزيد من البحث في هذا المجال.

# General Introduction

A multitude of industrial processes and applications across various sectors are powered by rotating electrical devices, which are essential components. In order to guarantee uninterrupted and efficient operations; where the reliable operation of these machinery is essential; ranging from power generation turbines to manufacturing assembly lines and transportation systems. The discovery of electromagnetic induction in the early 19th century and the pioneering work of Michael Faraday are the sources of the origins of rotating electrical devices. This facilitated the development of early generators and motors, which were instrumental in the electrification of modern society and the industrial revolution.   
 Nevertheless, rotating electrical machines are susceptible to a variety of defects and failures, which can result in severe repercussions if left undiagnosed and unaddressed, much like any intricate mechanical system. Faults in rotating electrical machines can manifest in a variety of ways, including mechanical failures such as bearing defects, rotor eccentricity, and shaft misalignment, as well as electrical failures such as stator winding insulation degradation, rotor bar breakages, and short circuits. These defects can result in costly downtime, production losses, and potential safety hazards, as well as increased energy consumption, excessive vibrations, and even complete system failure.

Historically, the detection and diagnosis of faults in rotating electrical devices have been reliant on vibration analysis techniques, periodic maintenance, and manual inspection. The origins of vibration monitoring can be traced back to the early 20th century, when researchers began to investigate the correlation between machine vibrations and potential defects. In order to analyse vibration signals and derive pertinent features for fault detection, a variety of signal processing techniques, including Fast Fourier Transform (FFT) and wavelet analysis, were developed over time.

Nevertheless, these conventional methods frequently necessitate a high level of domain expertise, are time-consuming, and may not identify incipient faults in their initial phases, resulting in the loss of critical opportunities for timely interventions and preventive maintenance. In response to the increasing demand for reliability and efficiency and the complexity of industrial processes, the constraints of manual defect detection methods became increasingly apparent.

In the past few years, the accelerated development of artificial intelligence (AI) and machine learning (ML) technologies has created new opportunities for the automation of fault detection and predictive maintenance workflows. AI and ML techniques have been widely adopted in a variety of domains, such as defect detection and diagnostics, as a result of the increasing availability of computational power and large datasets, as well as advancements in algorithms and neural network architectures.

In particular, deep learning, a subset of machine learning that replicates the hierarchical structure of the human brain, has demonstrated exceptional success in a variety of fields, such as signal analysis, natural language processing, and computer vision. The origins of deep learning can be traced back to the 1940s and 1950s, when artificial neural networks were developed, following the work of researchers such as Warren McCulloch and Walter Pitts. Nevertheless, it was not until the late 20th and early 21st centuries that deep learning gained significant traction. This was facilitated by the availability of large datasets, increased computational power, and algorithmic advancements, such as the introduction of convolutional neural networks (CNNs) and more effective training techniques.

**CHAPTER 1**

**Faults of rotating machines**

# Chapter 1

## Introduction:

The main advantage of vibration measurement on rotating machinery is the ability to detect faults before a failure occurs that causes an unplanned shutdown of a machine.

In this chapter, we review the components of the rotating machine and the faults that may occur. At the end, we discuss the diagnostic methods currently applied to the machine.

## components of rotating machine:

The rotating machine is composed of the main elements shown in Figure Ⅰ -1:

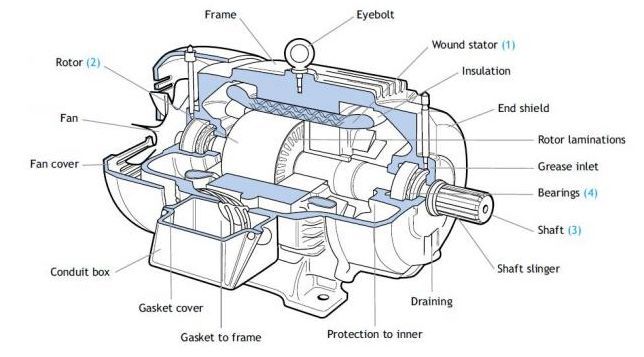
****

Figure ‎I‑1:Vue of rolling machine

* + 1. Stator:

The different types of rolling motors are distinguished only by the rotor in all cases, the stator remains the same in principle. It consists of a wound winding distributed in the slots of the stator magnetic circuit. This magnetic circuit consists of a stack of laminations in which slots parallel to the axis of the machine are cut out. [1][2]

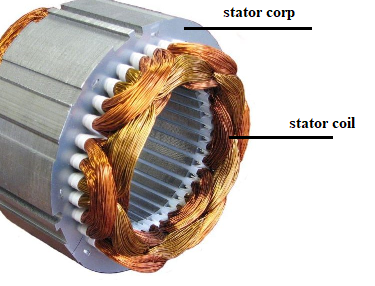


Figure ‎I‑2: stator of rolling machine

* + 1. Rotor:

The rotor is the component that rotates in an electrical machine. The same definition is valid whether the electric machine is an electric motor or an electric generator.

In an electric motor, the rotor works together with the stator (fixed part) to transmit the power of the electric machine.

In addition to being a component of an electric motor, the term is commonly used in rotating machines, such as turbines and centrifugal pumps, as opposed to the so-called

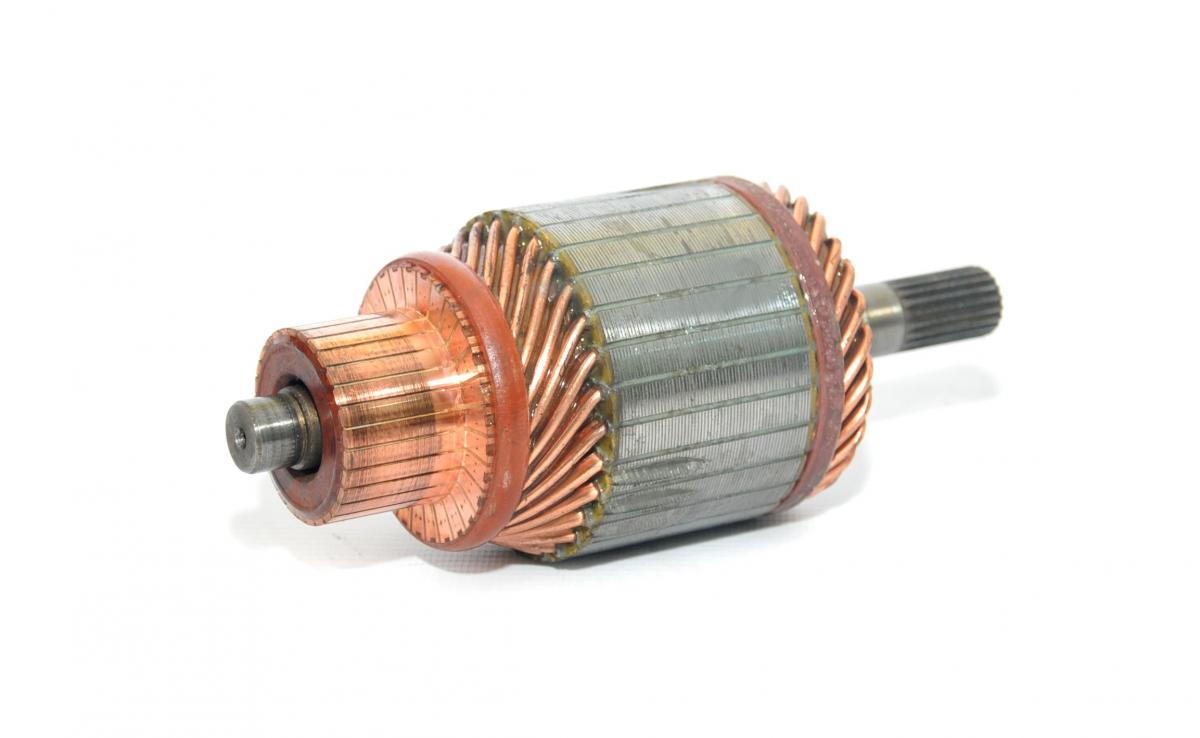
stator. [3]

Figure ‎I‑3: rotor of rolling machine

* + 1. Shaft:

A shaft is a rotating machine element, usually circular in cross-section, which is used to transmit power from one part to another, or from a machine that produces power to a machine that absorbs power.

* + 1. Bearings:

Rolling element bearings are widely used machine elements and they are used to allow rotation of or around shafts in many different machines, such as bicycles, cars, electric motors, aircraft turbines, mills, etc. The main advantage of rolling bearings is their low friction, since the resistance to the motion is only driven by the rolling friction which is much lower typically than the sliding friction. The design of the rolling-element bearings ensures their reliability while reducing the impact of variations in speed, temperature, load, and lubricating conditions. At the same time, the maintenance required for such bearings is typically much easier as compared to the other types of bearings. All these features coupled with the availability and the efficiency of the bearings contribute to a wide use of rolling element bearings in various applications. [3]

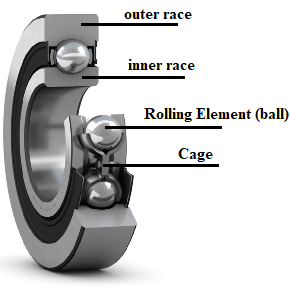


Figure ‎I‑4: rolling Element bearings

## Faults Study of electrical machines

**Definition :** Fault in electrical machines refers to any abnormal condition or deviation from the designed operating parameters that can lead to a partial or complete failure of the machine. Faults can be electrical, mechanical, or thermal in nature, and they can arise due to various reasons, such as insulation degradation, bearing wear, overloading, or manufacturing defects.

The study of faults in electrical machines is an essential aspect of electrical engineering, as it helps in understanding the potential failures, their causes, and the mitigation strategies to ensure reliable operation and maintenance of these machines. Electrical machines, such as motors and generators, are complex systems that can experience various types of faults during their operation. These faults can lead to decreased performance, increased energy consumption, and even complete system failure if not detected and addressed promptly.[4]

* + 1. Causes of faults:

The causes of electrical machine faults can be classified into: Electrical causes, Mechanical causes, Thermal cause, Environmental factors.

* + 1. Importance of fault study:

1. Reliability and availability:
   * Identifying and mitigating faults can prevent unexpected failures and downtime, improving the overall reliability and availability of electrical machines.
2. Safety and risk management:
   * Faults in electrical machines can pose safety risks to personnel and equipment. Fault analysis helps in implementing appropriate safety measures and risk mitigation strategies.
3. Maintenance optimization:
   * Understanding fault mechanisms and their progression enables the development of effective condition monitoring and predictive maintenance strategies, leading to optimized maintenance schedules and reduced operational costs.
4. Design improvements:

* Fault analysis provides valuable insights for designing fault-tolerant machines, improving the overall robustness and durability of electrical machines.

1. Energy efficiency:
   * Faults can lead to increased energy consumption and inefficient operation. Early fault detection and mitigation can help maintain optimal energy efficiency and reduce operational costs.
2. Root cause analysis:
   * Investigating the root causes of faults helps in identifying and addressing underlying issues, preventing recurrence, and enhancing the overall reliability of electrical machines.

The study of faults in electrical machines is a critical aspect of electrical engineering, as it contributes to the safe, reliable, and efficient operation of these essential components in various industries, including power generation, transportation, manufacturing, and automation.

* + 1. Classification of faults in electrical machines:

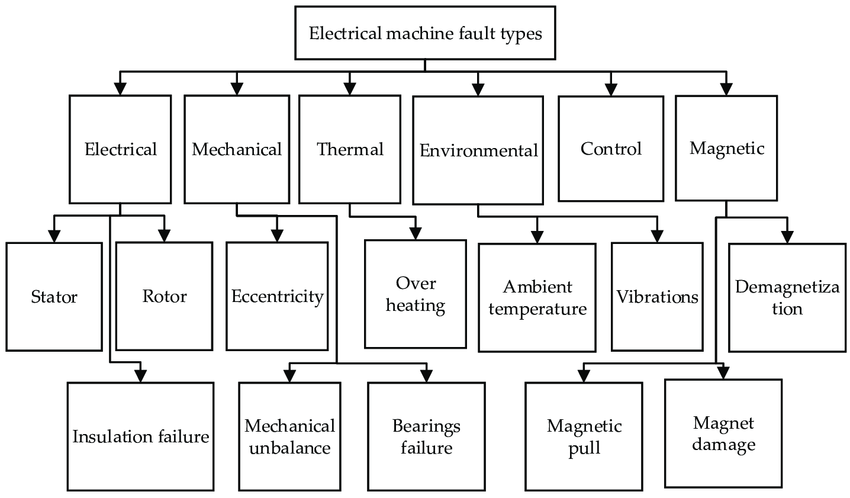
A classification of faults that exist in electrical machines according to their origins is presented in the Figure Ⅰ-5. [5]

Figure ‎I‑5 : Classification of defects according to their origins

The causes of defects in the stator and rotor are multiple, the most frequent of which are listed below:

* + - 1. Stator defects:
* Short circuit between turns: overvoltage, excessive temperature, vibration, humidity.
* Short circuit between phases: high temperature, unbalanced power supply, installation defect.
* Insulation defect: frequent starting, partial discharge, extreme condition, temperature and humidity.
* Defect between the stator and frame: thermal cycling, abrasion of the insulation, clogging of the windings by the frame, presence of sharp points in the slots, impact.
* Displacement of conductors: frequent starting, vibration of winding heads.
* Failure of connectors: excessive vibration.
* Frame vibration: poor installation, magnetic unbalance, unbalanced power supply, overload, winding movement, contact with the rotor.[5]
  + - 1. Rotor defects:
* Bearing defect: poor installation, magnetic unbalance, high temperature, loss of lubricant, unbalanced load, corrosion.
* Broken bars: thermal cycling, long duration transient operation; magnetic unbalance.
* Broken portion of ring: thermal cycling.
* Eccentricity: poor installation, magnetic unbalance, bearing defects.
* Bearing misalignment: coupling defect, poor installation, overload.
* Magnetic circuit defect: manufacturing defect, overload, thermal cycling.
* Mechanical unbalance: misalignment, movement of short-circuit rings.
* Defect due to the power supply network: The networks and electrical installations are subject to random incidents, the most frequent of which are:
* Short circuit between phases.
* Power supply phase outages.
* Unbalanced supply voltages. Page 8 The repercussions of these anomalies on service continuity and equipment operation depend on the nature of the defect. The latter is caused either in overhead networks or by constraints of a:
* climatic nature (rain, lightning, etc.).
* environmental nature (tree branches, lead weights, etc.).
* following the interconnection of different networks. It follows that electrical installations may undergo a number of voltage disturbances that are difficult to predict, characterized by either a transient voltage dip or a brief outage. In the most serious cases, this causes a long-term outage.[6]

## Fault due to the engine:

The failures that can affect the machine are of various origins: electrical, mechanical, and even magnetic.

* + 1. Mechanical Failures:
       1. Bearing Defects

Ball bearings play a very important role in the operation of all types of electrical machines.

Bearing defects have many causes such as fatigue spalling, lubricant contamination, excessive load, or electrical causes such as the circulation of leakage currents induced by inverters, rotation problems within the bearing housing caused by a damaged, scaled, or cracked winding can create disturbances within the machine, as electrical currents circulate at the level of the bearings of an asynchronous machine and for high speeds can lead to the deterioration of the latter.

* + - 1. Eccentricity Defects

Sometimes, the machine can be subjected to rotor eccentricity, resulting in torque oscillations (Space between the center of the rotor and the shaft's canter of rotation).

This phenomenon is called eccentricity, the origin of which can be related to incorrect positioning of the bearings during assembly, a bearing defect (wear), a load defect, or a manufacturing defect (machining).

1. **Static eccentricity**: is generally due to a misalignment of the rotor's axis of rotation with respect to the stator's axis.
2. **Dynamic eccentricity:** Dynamic eccentricity occurs when the center of rotation of the rotor is different from the geometric center of the stator, but in addition, the center of the rotor rotates around the geometric center of the stator. This type of eccentricity is caused by a deformation of the rotor or stator cylinder.[7]
3. **Mixed eccentricity**: represents the sum of the static and dynamic cases.
   * 1. Electrical Failures:
        1. Stator Faults:

The occurrence of a fault in the stator electrical circuits of the asynchronous machine can have various origins. For example, short circuits between turns of the same phase are a common fault that can occur either at the coil heads or in the slots. This type of fault can be caused by insulation degradation of the stator winding turns. Other examples include short circuits between a phase and the neutral, between a phase and the machine's metal frame, or between two stator phases.

* + - 1. Rotor Faults:

Analyzing the spectrum of the stator current in steady-state provides indications of rotor faults such as bar breaks and short-circuit rings.

* + - 1. Bar Breakage Fault:

The breakage of bars in an asynchronous machine is one of the most commonly studied faults in the laboratory due to its simplicity. Bar breakage causes rotor asymmetry, resulting in the creation of a rotating field opposite to that generated by the stator, at the slip frequency. This leads to the generation of an additional current in the stator winding.

* + - 1. Breakage of a Portion of the Short-Circuit Ring:

The breakage of a portion of the short-circuit ring is a fault that occurs as frequently as bar breakage. These breaks are caused by casting bubbles or differential expansions between the bars and the rings, especially since the short-circuit ring portions carry higher currents than the rotor bars. This fault is challenging to detect and is often grouped or confused with bar breakage in statistical studies. Therefore, improper ring sizing, deteriorating operating conditions (temperature, humidity), or an overload of torque and currents can lead to their breakage.

## Conclusion

Rotating electrical machines are important parts of many industries, and they need to work reliably to keep things running smoothly and avoid downtime. This chapter gave a thorough look at the different parts of rotating machines, the problems that can happen, and the reasons behind them. For proper fault diagnosis and maintenance, it is important to know the different types of faults, such as mechanical failures (such as bearing defects and eccentricity) and electrical failures (such as stator and rotor faults), as well as what causes them. Electrical faults like stator winding insulation degradation, rotor bar breakages, a ring faults can cause performance loss and even failure. Mechanical faults like bearing defects and eccentricity can cause vibrations, noise, and more wear. Finding and diagnosing these faults as soon as possible is important for putting in place the right mitigation strategies, such as predictive maintenance, condition monitoring, and repairs or replacements that are done on time. By looking into how these faults behave and what they look like, engineers and maintenance workers can come up with better ways to find faults, make maintenance schedules more efficient, and make rotating machines last longer. Also, the information gathered from fault analysis can be used to improve the design and production of these machines, making them more reliable, using less energy, and working better overall. To keep rotating electrical machines safe, efficient, and long-lasting in a variety of settings, more research and improvements must be made in fault diagnosis and condition monitoring.

**Chapter 02:**

**Deep learning**

# Chapter 2



## Deep Learning

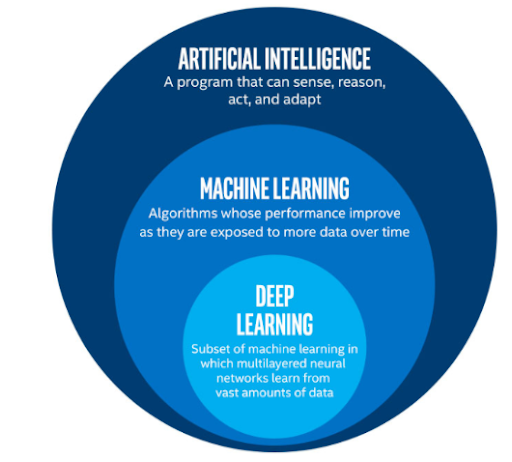
Within the field of artificial intelligence, deep learning is a subset of machine learning. Synonyms for "artificial intelligence" include methods that make computers behave like people. As a group of algorithms trained on data, machine learning makes all of this possible. The structure of the human brain inspired deep learning, which is just a type of machine learning.as illustrated in Figure Ⅱ-1. [8]

Figure ‎II‑1:AI vs. machine learning vs. deep learning

## Deep Learning function

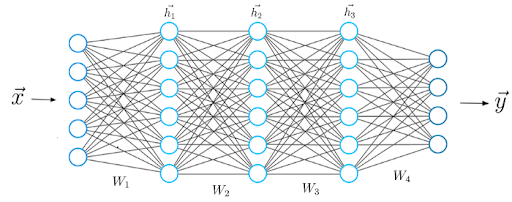
By consistently analyzing data with a predetermined logical structure, deep learning algorithms endeavor to arrive at conclusions that are comparable to those of humans. Deep learning employs a multi-layered structure of algorithms known as neural networks to accomplish this.

Figure ‎II‑2:A typical neural network

The neural network's design is predicated on the structure of the human brain. In the same way that our brains are capable of identifying patterns and classifying various types of information, it is possible to train neural networks to perform the same tasks on data.

The individual layers of neural networks can also be viewed as a type of filter that operates from the most obvious to the subtlest, thereby increasing the probability of detecting and generating an accurate result. The human brain functions in a similar manner. The brain attempts to compare new information with known objects whenever it is received. Deep neural networks also employ the identical concept.

Neural networks enable us to perform many tasks, such as clustering, classification or regression.

Unlabelled data can be grouped or sorted based on the similarities between samples in the data using neural networks. Alternatively, in the context of classification, the network can be trained on a labelled data set to categorize the samples in the data set into distinct categories.

In general, neural networks are capable of executing the same tasks as standard machine learning methods (but classical algorithms cannot perform the same tasks as neural networks). In other words, artificial neural networks include distinctive features that allow deep learning models to resolve problems that machine learning models are incapable of resolving.

Deep learning has been the driving force behind all of the recent advancements in artificial intelligence. Self-driving cars, chatbots, and personal assistants such as Siri and Alexa would not exist without deep learning. Netflix would have no idea which movies to recommend, and Google Translate would remain as rudimentary as it was prior to Google's transition to neural networks. All of these deep learning applications and technologies are powered by neural networks.

Artificial neural networks and deep learning are the driving forces behind a new industrial revolution. Ultimately, deep learning is the most straightforward and effective method of achieving genuine machine intelligence that we have ever encountered. [8]

## The prevalence of deep learning

The potential for enhanced accuracy through the utilization of big data and the absence of the need for feature extraction are among the reasons for the widespread use of deep learning.

* + 1. ****No Feature Extraction****

We employed conventional machine learning techniques, such as decision trees, SVM, naïve Bayes classifiers, and logistic regression, for an extended period prior to the implementation of deep learning. These algorithms are also referred to as flat algorithms. The term "flat" in this context denotes that these methods are typically unable to be directly applied to the raw data (such as .csv, images, text, etc.). A preprocessing procedure known as feature extraction is required.

The output of feature extraction is a representation of the raw data that these traditional machine learning algorithms can employ to execute a task. For instance, the data can now be categorized into multiple categories or classes. Feature extraction is typically highly intricate and necessitates a comprehensive understanding of the problem domain. In order to get optimal outcomes, this preprocessing layer must be modified, tested, and polished across a number of repetitions.

The feature extraction stage is unnecessary for artificial neural networks that are trained with deep learning. The layers are capable of directly and independently acquiring an implicit representation of the raw input.

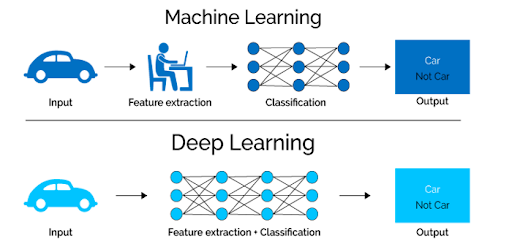
The process is as follows: Over the course of numerous layers of an artificial neural network, the raw data is represented in a compressed and abstract manner. Next, we employ this compressed version of the input data to generate the outcome. A possible outcome is the categorization of the incoming data into distinct categories.

Figure ‎II‑3:deference between deep learning and ML algorithms

In order to achieve the most accurate abstract representation of the input data, our neural network optimizes this phase throughout the training process. This implies that little to no manual effort is necessary to complete and optimize the feature extraction process for deep learning models.

Allow us to examine a specific instance. When employing a machine learning model to ascertain whether a specific image depicts a car or not, it is necessary for humans to first identify the distinctive characteristics of a car, such as its shape, size, windows, and wheels. Subsequently, the feature is extracted and provided to the algorithm as input data. By doing so, the algorithm would classify the photographs. That is, in order for the model to reach a conclusion, a programmer must directly intervene in the process of machine learning.

The phase of feature extraction is entirely superfluous in the context of a deep learning model. Without human interaction, the program would automatically identify these distinctive attributes of a vehicle and generate accurate forecasts.

Indeed, the prohibition against collecting data features is applicable to any other neural network-related tasks undertaken. Simply provide the neural network with the raw data, and the model will handle the rest. [8]

* + 1. Big Data improves deep learning accuracy

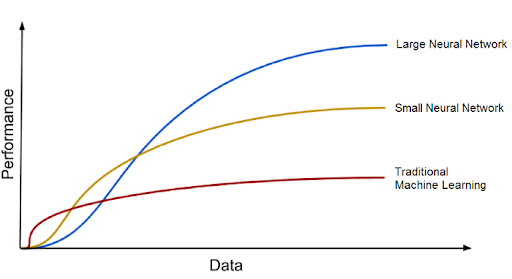
The second significant advantage of deep learning, and a critical component of comprehending its increasing popularity, is that it is powered by vast quantities of data. The era of big data will offer significant opportunities for the development of innovations in deep learning.

Figure ‎II‑4:Deep learning algorithms improve with increasing amounts of data.

After a saturation point, conventional machine learning models, such the naive Bayes classifier and SVM, stop to progress, but deep learning algorithms tend to enhance their accuracy as the quantity of training data rises. [8]

## Neural Networks in Machine Learning

For object detection and image processing, fully connected neural networks another term for classical neural networks are employed. Classical neural networks consist of multilayer perceptron’s with neurons connected to a continuous network. These consist of the three functions that are used to convert the model into primary binary data, specifically:

* **Linear function:** has a single line that multiplies its inputs by a constant multiplier.
* **Non-linear function:** It is further divided into three subsets:
* **Sigmoid curve**: An S-shaped curve with a range of 0 to 1.
* **Hyperbolic Tangent**: An S-shaped curve that ranges from -1 to 1.
* **Rectified Linear Unit (ReLU):** A single point function returns 0 when the input value is less than the set value and returns a linear multiple if the given input is greater than the set value.[8]

## Convolutional Neural Network (CNN) in deep Learning

A Convolutional Neural Network (CNN) is a deep learning algorithm that is uniquely well-suited for image recognition and processing tasks. It comprises numerous layers, such as fully connected layers, convolutional layers, and pooling layers. CNNs are well-suited for capturing hierarchical patterns and spatial dependencies within images, as their architecture is inspired by the visual processing in the human brain. [9]

A Convolutional Neural Network is composed of the following key components:

1. **Convolutional Layers:** These layers employ filters (sometimes referred to as kernels) to identify characteristics like as edges, textures, and more intricate patterns by applying convolutional operations to input pictures. The spatial associations between pixels are preserved by convolutional procedures.
2. **Pooling Layers:** The computational complexity and the number of parameters in the network are reduced by down sampling the spatial dimensions of the input in pooling layers. Max pooling is a frequently performed pooling method that involves selecting the highest value from a collection of neighbouring pixels**.**
3. **Activation Functions:** The model is able to learn more complex associations in the data by including non-linearity, such as the Rectified Linear Unit (ReLU).
4. **Fully Connected Layers**: These layers are responsible for generating predictions by utilizing the high-level features that were acquired by the preceding layers. They establish connections between each neuron in one layer and each neuron in the subsequent layer.

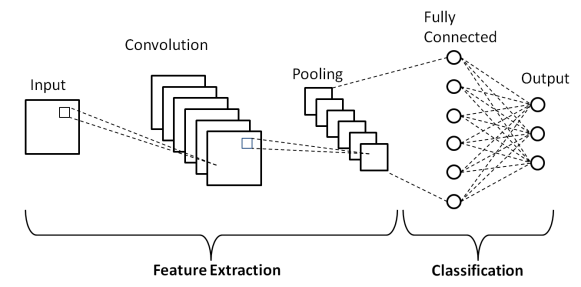


Figure ‎II‑5design convolution neural network

To enable CNNs to recognize patterns and attributes associated with certain items or classes, they are trained using a substantial dataset of labelled pictures. Achieved cutting-edge outcomes in a variety of computer vision applications, demonstrating extraordinary efficacy in image-related tasks. The ability to automatically learn hierarchical representations of characteristics renders them particularly well-suited for scenarios in which spatial relationships and patterns in the data are critical for accurate prediction. Image classification, item identification, facial recognition, and medical image analysis are among the several applications in which CNNs are employed.

Elements such as edges, textures, and shapes are extracted from the input image by applying filters in the convolutional layers, which are the fundamental component of a CNN.

The feature maps are down-sampled to reduce the spatial dimensions while retaining the most pertinent data after the convolutional layers' output is processed through pooling layers. To identify or forecast the image, one or more fully connected layers are subsequently applied to the output of the pooling layers. [9]

* + 1. Convolutional Neural Network Design
* A convolutional neural network is a multi-layered feed-forward neural network that is constructed by stacking numerous unseen layers in a specific order.
* The sequential design of CNN enables it to acquire hierarchical features.
* Convolutional layers are often followed by activation layers in CNNs, with some of them being followed by grouping layers and hidden layers.
* The pre-processing required in a ConvNet is analogous to the pattern of neurons in the human brain and was inspired by the organization of the Visual Cortex. [9]
  + 1. Convolutional Neural Network Training

A supervised learning technique is employed to train CNNs. In other words, the CNN is provided with a sequence of annotated training images. CNN subsequently develops the capacity to precisely classify the photographs that are submitted.

The subsequent steps comprise the training procedure for a CNN:

1. **Data Preparation:** The training photos are preprocessed to guarantee that they are same in terms of size and format.
2. **Loss Function:** A loss function is employed to evaluate the effectiveness of the CNN on the training data. The loss function is commonly determined by subtracting the anticipated labels from the actual labels of the training pictures.
3. **Optimizer:** A CNN's weights are updated by an optimizer to reduce the loss function.
4. **Backpropagation:** is a method that is employed to determine the gradients of the loss function in relation to the weights of the CNN. Subsequently, the optimizer updates the weights of the CNN based on the gradients.
   * 1. CNN Evaluation

Following training, CNN can be evaluated on a held-out test set. The test set is a collection of images that the CNN has not encountered throughout the training process. The performance of a CNN on the test set is a trustworthy predictor of its ability to function with real data.

A variety of parameters can be used to evaluate the performance of a CNN on photo categorization tasks. The metrics that are most frequently employed are as follows:

* **Accuracy:** The percentage of test pictures that the CNN properly identifies is known as accuracy.
* **Precision:** isthe proportion of test images that the CNN predicts as belonging to a specific class and that actually belong to that class.
* **Recall**: Recall is the proportion of test images that belong to a specific class and that the CNN predicts as belonging to that class.
* **F1 Score:** The F1 Score is a harmonic mean of precision and recall. It is an effective statistic for assessing the performance of a CNN in imbalanced classes. [10]
  + 1. Applications of CNN
* **Image classification:** The most advanced models for classifying images are CNNs. Images can be categorized using them into groups like vehicles and trucks, dogs and cats, and flowers and animals.
* **Object detection:** CNNs are useful for identifying individuals, vehicles, and structures in pictures. They can also be used to locate an object in an image by using a technique called object localization.
* **Image segmentation:** CNNs can recognize and label various items in an image by using image segmentation techniques. Applications like robotics and medical imaging can benefit from this.
* **Video analysis: CNNs** can be used to evaluate videos in a variety of ways, including monitoring objects and identifying events. Applications like traffic monitoring and video surveillance can benefit from this.[10]
  + 1. Advantages of CNN
* Cutting-edge accuracy can be achieved using CNNs on a variety of image recognition tasks, such as object identification, picture segmentation, and image classification.
* CNNs have the potential to be highly effective, especially when utilized in conjunction with specialized equipment such as GPUs.
* CNNs are capable of withstanding a certain degree of data volatility and noise in the input.
* By modifying the network's design, CNNs can be programmed to do a diverse array of jobs.
  + 1. Disadvantages of CNN
* CNNs can be difficult to train, especially when working with large datasets, because to their complexity.
* The training and deployment of CNNs may necessitate a substantial amount of computing power.
* CNNs necessitate an abundance of tagged data for training.
* Interpreting CNNs can be difficult, which means it is difficult to understand the reasoning behind the predictions they create. [10]

## Conclusion:

In this chapter, we have explored the fundamentals of deep learning and convolutional neural networks (CNNs), which are powerful tools for machine learning, particularly in computer vision tasks. We have discussed the key components of CNNs, including convolutional layers, pooling layers, activation functions, and fully connected layers, and how they work together to learn hierarchical representations of features from input data.

We have also delved into the design, training, and evaluation processes of CNNs, highlighting the importance of techniques such as data preparation, loss functions, optimization algorithms, and backpropagation in achieving accurate and robust models.

Furthermore, we have explored the various applications of CNNs, ranging from image classification and object detection to image segmentation and video analysis, showcasing their versatility and effectiveness in diverse domains.

While CNNs offer numerous advantages, including cutting-edge accuracy, computational efficiency, and flexibility, we have also acknowledged their potential drawbacks, such as the complexity of training, computational resource requirements, and the need for large labeled datasets.

As deep learning and computer vision continue to evolve, it is essential to stay updated with the latest advancements in CNN architectures, training techniques, and optimization strategies. The knowledge gained in this chapter serves as a solid foundation for further exploration and practical implementation of CNNs in real-world applications.

**Chapter 03**

**Case study and Simulation Results**

# Chapter 3



## Introduction

In this part, we will go through the machine fault detection using deep learning. We first present the used database and the preprocessing of the signal to be used in the next step of detection.

## Fault detection data base

Experiments are developed to evaluate the efficacy of the proposed 2d-CNN model by employing a bearing dataset from the Bearing Data Centre at Case Western Reserve University (CWRU).

The accelerometer of a motor-driven mechanical system was used to acquire the original experimental data at a sampling frequency of 12 kHz. Figure ‎III.1 illustrates the test stand. The rotating speed of the bearing and the sampling frequency of the sensor can be used to infer the number of data points collected during a single rotation of the bearing. The functional relationship between the two can be expressed as (4), where n is the number of data points collected per circle, f is the sampling frequency, and v is the rotating speed (rpm):

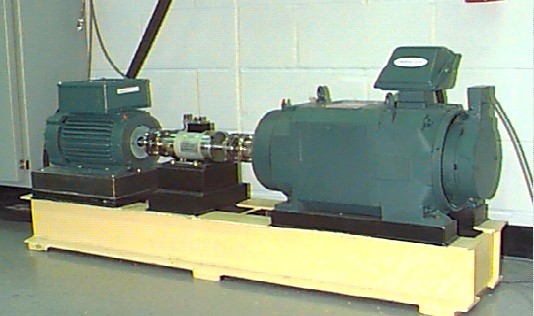
 *n* = *f* ∗ 60/*v*, (4)

Figure ‎III‑1:Test stand for collecting vibration signals from REBs used by CWRU.

The damage of the bearings utilized in the experiment is initiated by electro-discharge machining (EDM), as illustrated in Figure ‎III.2. The inner ring, outer ring, and rolling element are the three potential locations for bearing failure. We collected only two fault sizes for each location, 7 mils and 21 mils, for the sake of convenience in this study. Consequently, our experiments contain six distinct signal types, including the normal signals and the faulty ones. [11]

Figure ‎III‑2: Photographs of three kinds of damaged bearings



The 2dCNN model is susceptible to overfitting in the absence of sufficient training data, as it is required to acquire a substantial number of parameters and is only capable of functioning with a substantial number of labeled samples.

Consequently, the data augmentation approach is implemented, which involves the addition of the subsequent n sampling points to the current sample and the removal of the previous n sampling points to generate a new sample.

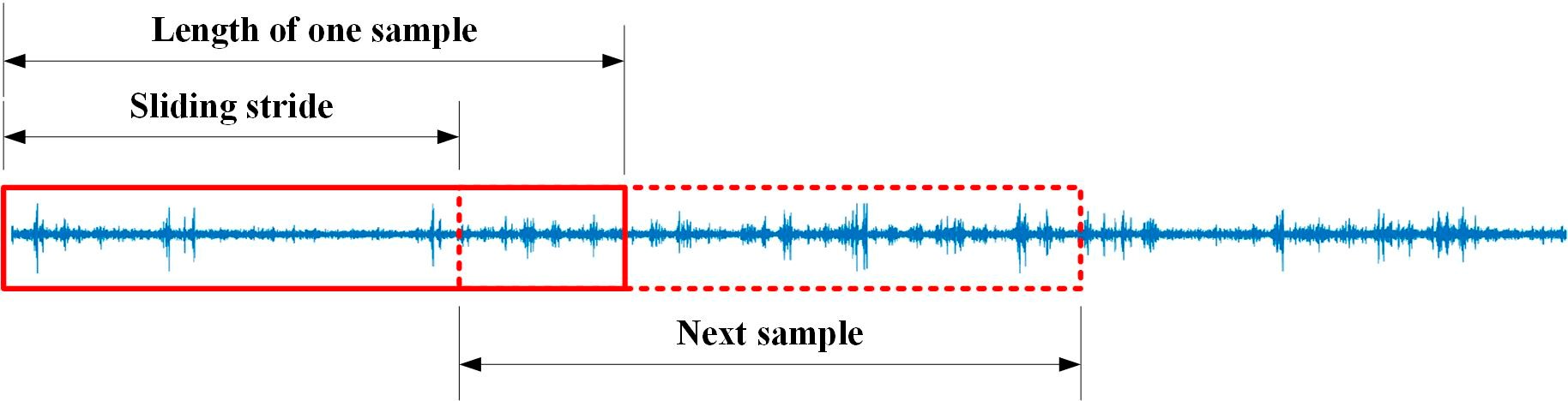


Figure ‎III‑3: Diagram of the data augmentation.

In our experiments, a single sample is composed of 1024 sampling points or 4096 sampling points, contingent upon the size of the image (32\*32 or 64\*64). The sliding stride is contingent upon the total length of the vibration signal and the number of samples needed when the length of one sample is fixed.

The comprehensive descriptions of the experimental datasets are provided in Table (1) and Table (2). The time domain waveforms of the normal signal, B fault signals (7, 21), IR fault signals (7, 21), and OR fault signals (7, 21), as well as bearing vibration signals, can be observed in Figures (‎III ‑4, ‎III‑5).

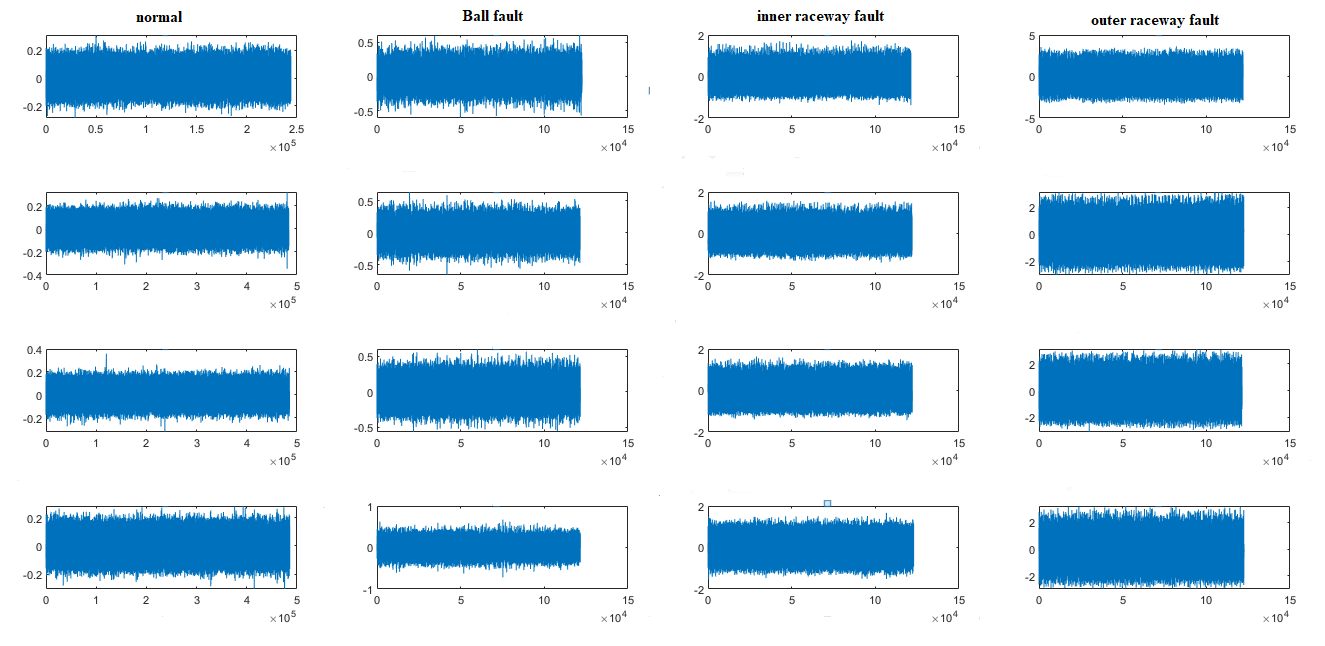
****

Figure ‎III‑4:Time domain waveforms of bearing vibration signals (7mm) in 4 different speeds

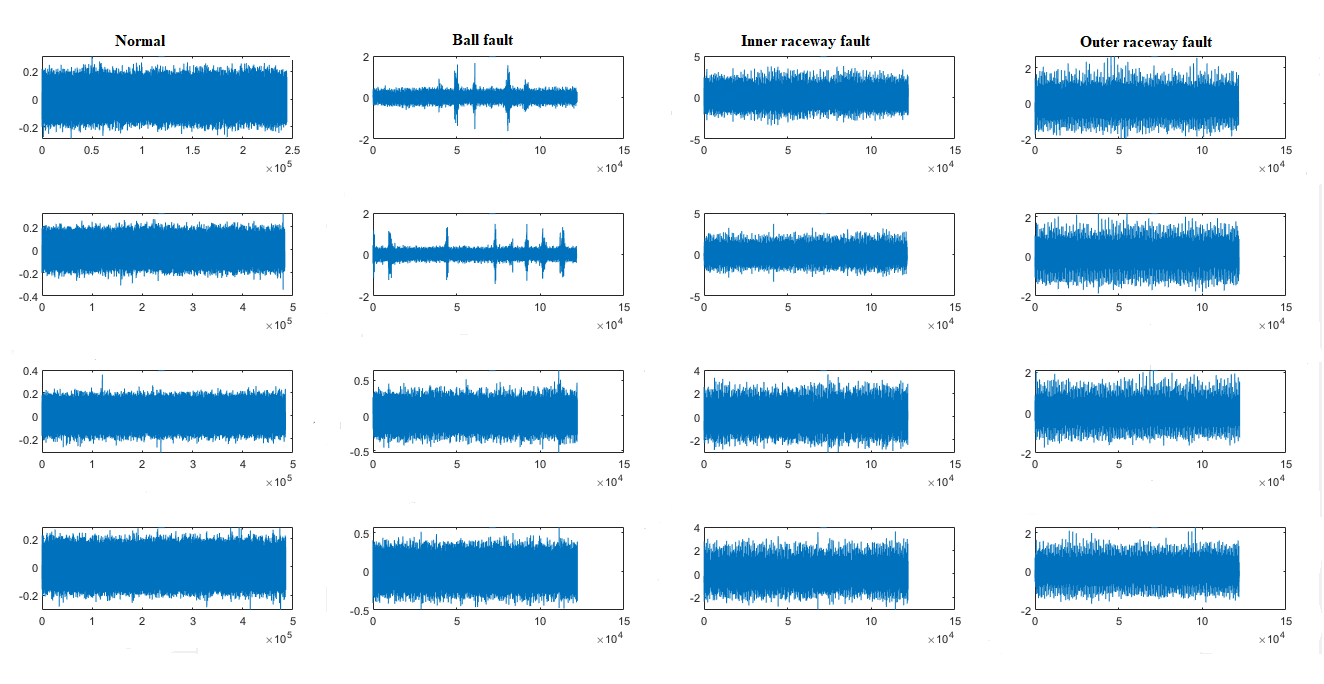
****

Figure ‎III‑5:Time domain waveforms of bearing vibration signals (21mm) in 4 different speeds

## The Structure of work:

The subsequent step is to establish the specific implementation procedure in MATLAB 2019a after the introduction of the datasets that will be used to train and evaluate the 2D CNN model.   
However, it is imperative to recognize that the Deep Learning Toolbox is not inherently included in the MATLAB 2019a version that is currently in use. The Deep Learning Toolbox provides pre-built architectures and high-level functions for the training of deep neural networks, including CNNs.

The data must be converted and represented as image files (e.g., PNG, JPEG) before it can be processed by MATLAB 2019, as this toolbox is unavailable. MATLAB 2019a is unable to directly interact with raw data formats that are frequently employed in deep learning datasets, necessitating this additional conversion step.

The data samples can be read and processed by the MATLAB code as input for the 2D CNN model implementation from inception after being converted to image file formats. This is achieved by employing the low-level functions that are included in MATLAB 2019a.

The following figures illustrate how this signal-to-image conversion approach enables the use of techniques such as 2D convolutional neural networks, which can automatically learn relevant patterns and features directly from the image data during training, without the need for manual feature engineering on the original 1D signals. (Figure III 6 and Figure III 7).

MATLAB (Matrix Laboratory) is a numerical computing environment and high-level programming language that is extensively employed in a variety of scientific and engineering disciplines. It is especially well-suited for the development of algorithms, data analysis, matrix and array operations, and visualization. A vast library of built-in functions and toolboxes for various domains, including signal processing, control systems, machine learning, and image processing, is provided by MATLAB, in addition to an interactive environment. MATLAB, a powerful tool for rapidly prototyping, analyzing data, and developing models and applications, is a user-friendly interface that integrates computation, visualization, and programming. It was developed by MathWorks.

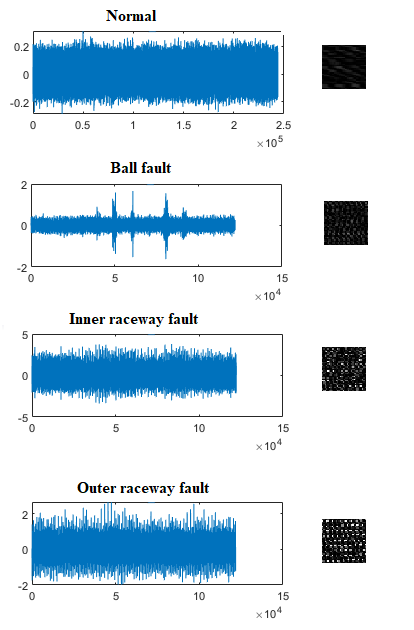
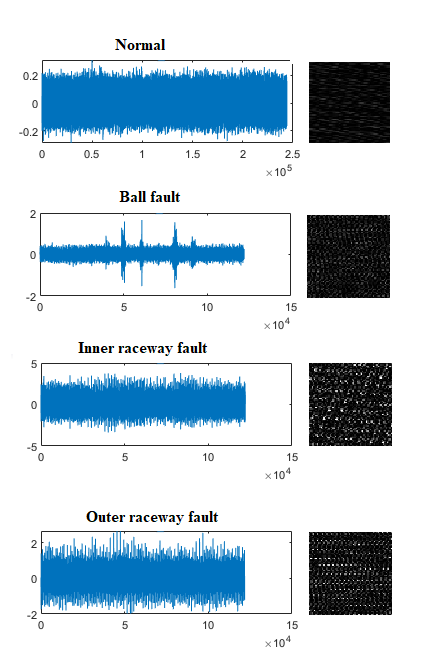
****

Figure ‎III‑6:signal-to-image conversion 64×64 pixel

Figure ‎III‑7signal-to-image conversion 32×32 pixel

The conversion of signals to images is illustrated in the two figures, which will be used to transform our database to images.

Therefore, we can provide a comprehensive description of the entire new data set in the table below.

**Table 1**:Description of all experimental datasets of 32×32 data size.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Fault Location** | **Normal** | **Ball** | | **Inner Ring** | | **Outer Ring** | | **Load (hp)** | **Speed (rpm)** |
| **Fault diameter (mil)** | 0 | 7 | 21 | 7 | 21 | 7 | 21 |  |  |
| **Dataset A** | | | | | | | |  |  |
| Train | 238 | 113 | 119 | 118 | 119 | 357 | 357 | 0 | 1797 |
| Test | 10% | | | | | | |  |  |
| **Dataset B** | | | | | | | |  |  |
| Train | 472 | 118 | 118 | 119 | 118 | 354 | 357 | 1 | 1772 |
| Test | 10% | | | | | | |  |  |
| **Dataset C** | | | | | | |  |  |  |
| Train | 473 | 117 | 120 | 119 | 118 | 354 | 354 | 2 | 1750 |
| Test | 10% | | | | | | |  |  |
| **Dataset D** | | | | | | | |  |  |
| Train | 474 | 118 | 119 | 120 | 119 | 354 | 354 | 3 | 1730 |
| Test | 10% | | | | | | |  |  |

**Table 2**:details about outer rigs dataset 32×32

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Outer Ring dataset** | **OR centred** | | **OR opposite** | | **OR orthogonal** | |
| 7 | 21 | 7 | 21 | 7 | 21 |
| **Dataset A** | 119 | 118 | 118 | 119 | 118 | 119 |
| **Dataset B** | 119 | 119 | 119 | 118 | 118 | 119 |
| **Dataset C** | 119 | 119 | 119 | 119 | 118 | 118 |
| **Dataset D** | 118 | 119 | 118 | 118 | 118 | 119 |

**Table 3**:Description of all experimental datasets of 64×64 data size

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Fault Location** | **Normal** | **Ball** | | **Inner Ring** | | **Outer Ring** | | **Load (hp)** | **Speed (rpm)** |
| **Fault diameter (mil)** | 0 | 7 | 21 | 7 | 21 | 7 | 21 |  |  |
| **Dataset A** | | | | | | | |  |  |
| **Train** | 59 | 29 | 29 | 29 | 29 | 87 | 87 | 0 | 1797 |
| **Test** | 10% | | | | | | |  |  |
| **Dataset B** | | | | | | | |  |  |
| **Train** | 118 | 29 | 29 | 29 | 29 | 87 | 87 | 1 | 1772 |
| **Test** | 10% | | | | | | |  |  |
| **Dataset c** | | | | | | |  |  |  |
| **Train** | 118 | 29 | 29 | 29 | 29 | 87 | 87 | 2 | 1750 |
| **Test** | 10% | | | | | | |  |  |
| **Dataset D** | | | | | | | |  |  |
| **Train** | 118 | 29 | 29 | 29 | 29 | 87 | 87 | 3 | 1730 |
| **Test** | 10% | | | | | | |  |  |

**Table 4**:details about outer ring dataset 64×64

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Outer Ring dataset** | **OR centred** | | **OR opposite** | | **OR orthogonal** | |
| 7 | 21 | 7 | 21 | 7 | 21 |
| **Dataset A** | 29 | 29 | 29 | 29 | 29 | 29 |
| **Dataset B** | 29 | 29 | 29 | 29 | 29 | 29 |
| **Dataset C** | 29 | 29 | 29 | 29 | 29 | 29 |
| **Dataset D** | 29 | 29 | 29 | 29 | 29 | 29 |

## Optimization Algorithms:

Training deep neural networks like convolutional neural networks (CNNs) involves minimizing a loss function by iteratively updating the model's weights and biases. The choice of optimization algorithm plays a crucial role in the efficiency and effectiveness of this training process. In this study, we explore two widely adopted optimization techniques:

* + 1. Stochastic Gradient Descent (SGD):

SGD is a classical and fundamental algorithm that updates the model parameters in the direction of the negative gradient of the loss function. It computes the gradients using a small subset (mini-batch) of training data at each iteration, making it computationally efficient for large datasets. However, SGD's constant learning rate and sensitivity to initialization can sometimes lead to suboptimal convergence.

* + 1. Adam (Adaptive Moment Estimation):

Adam is a more recent optimization algorithm that combines momentum and root-mean-square (RMS) propagation. It adaptively computes individual learning rates for each parameter based on the first and second moments of the gradients. Adam often exhibits faster convergence and better performance on sparse gradients compared to traditional SGD, making it a popular choice for training deep neural networks.

In this work, we comprehensively evaluate the performance of our 2D CNN model for image classification using both SGD and Adam optimizers. The Adam (Adaptive Moment Estimation) optimizer is our primary focus, as it has exhibited significant advantages over SGD in the training of deep networks.

The structure parameters and performance indicators employed in the deep learning model are delineated in the table below.

**Table 5**:structure parameters and performance indicator

|  |  |  |  |
| --- | --- | --- | --- |
| **Model** | **2d-DCNN** | | |
|  | **32**×**32** |  | **64×64** |
| Input layer | 32 × 32 |  | 64 × 64 |
| Layer-1 | Conv2d (5 × 5, 16) |  | Conv2d (5 × 5, 16) |
| Layer-2 | Pooling (2 × 2, 1) |  | Pooling (2 × 2, 1) |
| Layer-3 | Conv2d (5 × 5, 16) |  | Conv2d (5 × 5, 16) |
| Layer-4 | Pooling (2 × 2, 1) |  | Pooling (2 × 2, 1) |
| Layer-5 | Conv2d (3 ×3, 8) |  | Conv2d (5 × 5, 16) |
| Layer-6 | Pooling (2 × 2, 2) |  | Pooling (2 × 2, 2) |
| Layer-7 | Flatten |  | Flatten |
| Layer-8 | Dense (100) |  | Dense (100) |
| Layer-9 | Dense (10) |  | Dense (10) |
| **Mean accuracy** | **0.992** |  | **0.984** |

## Experiment and results

After implementing the 2D CNN architecture and training it using the Adam optimization algorithm, it is crucial to evaluate the model's performance quantitatively. This section presents the training results and a detailed analysis of the confusion matrix, providing insights into the model's classification capabilities.

The training process involved feeding the CNN with the image dataset, allowing it to learn discriminative features and patterns through multiple iterations. By monitoring the training and validation loss curves, we can assess the model's convergence behavior and detect any potential overfitting or underfitting issues.

In addition to the loss metrics, we report the overall classification accuracy achieved by the trained model on the held-out test dataset. This accuracy score serves as a reliable indicator of the model's ability to generalize and make correct predictions on previously unseen data samples.

* + 1. Case01 (SGD):
* **Input image data set 32×32**

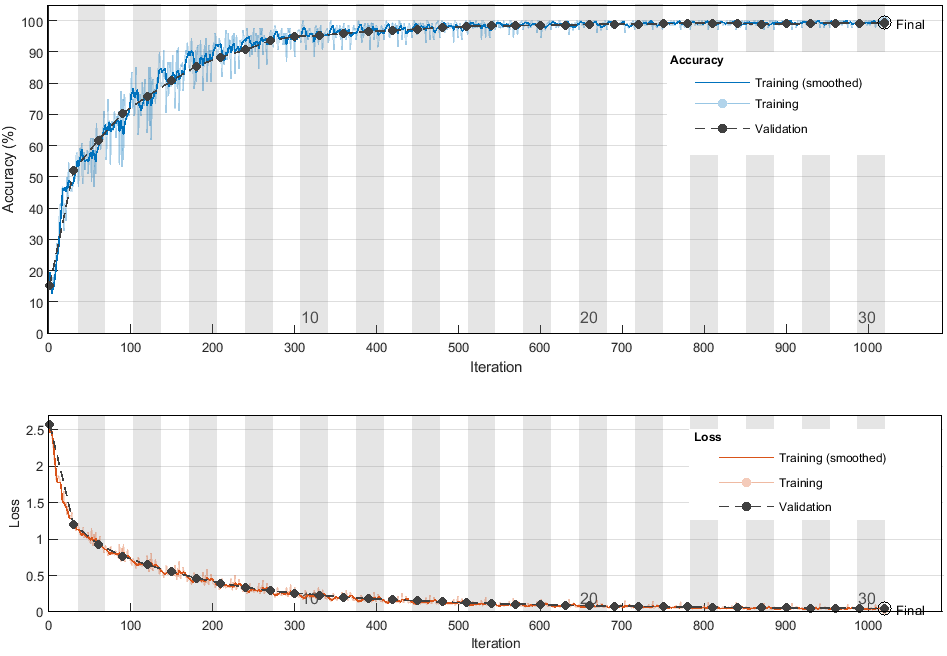
****

Figure ‎III‑8:Diagram of the 2d-DCNN model the training (SGD 32×32)

The image displays training plots for a machine learning model. The top plot shows the accuracy curves over training iterations for the training, smoothed training, and validation datasets. The validation accuracy reaches around 98.83% by the end of training. The bottom plot depicts the corresponding loss curves, which steadily decrease as training progresses. The model was trained for 30 epochs using a piecewise learning rate schedule with a learning rate of 0.0001, over approximately 1020 iterations. The plots suggest the training converged successfully, achieving high validation accuracy and low loss, indicating good performance on the given task.

#### Input image data size (64×64)

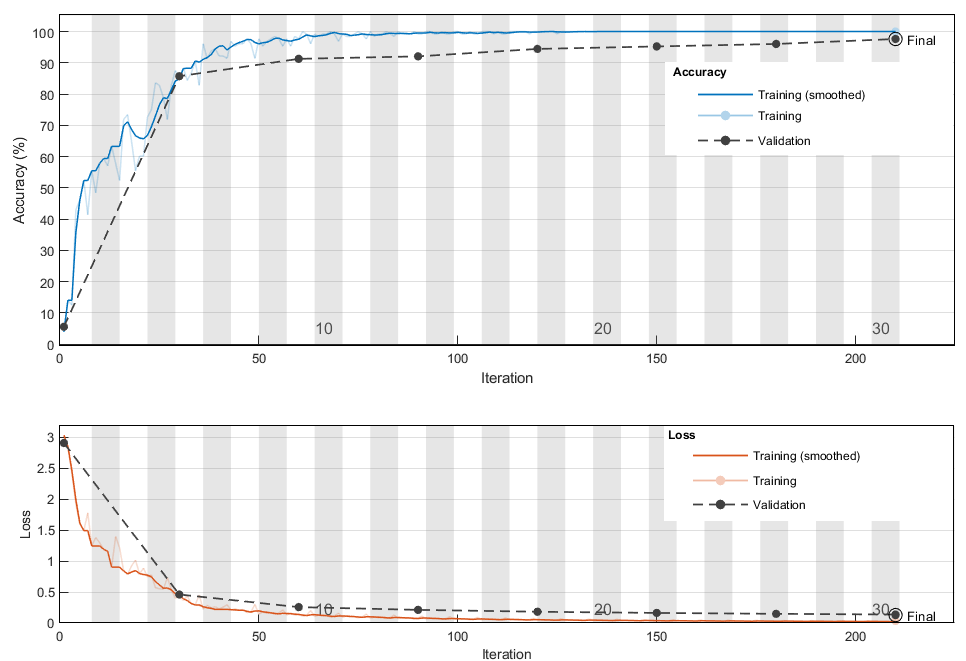
****

Figure ‎III‑9:Diagram of the 2d-DCNN model the training (SGD 64×64)

The training overview shows the model achieved a validation accuracy of 97.62% after 30 epochs of training. The learning rate was set to 0.0001 using a piecewise schedule, which likely involved adjusting the learning rate at certain intervals during training. Despite training for only 210 iterations, which is relatively low compared to the previous example of 1020 iterations (32\*32 pixels), the model still managed to reach a reasonably high validation accuracy close to 98%. This suggests the model architecture and training process were efficient and effective for the given task. The piecewise learning rate schedule likely helped the training converge faster by adapting the learning rate appropriately. Overall, these results demonstrate successful model training with a high final validation accuracy attained within a modest number of training iterations.

* + 1. Case 02 Adam:

#### input image data a size (32×32)

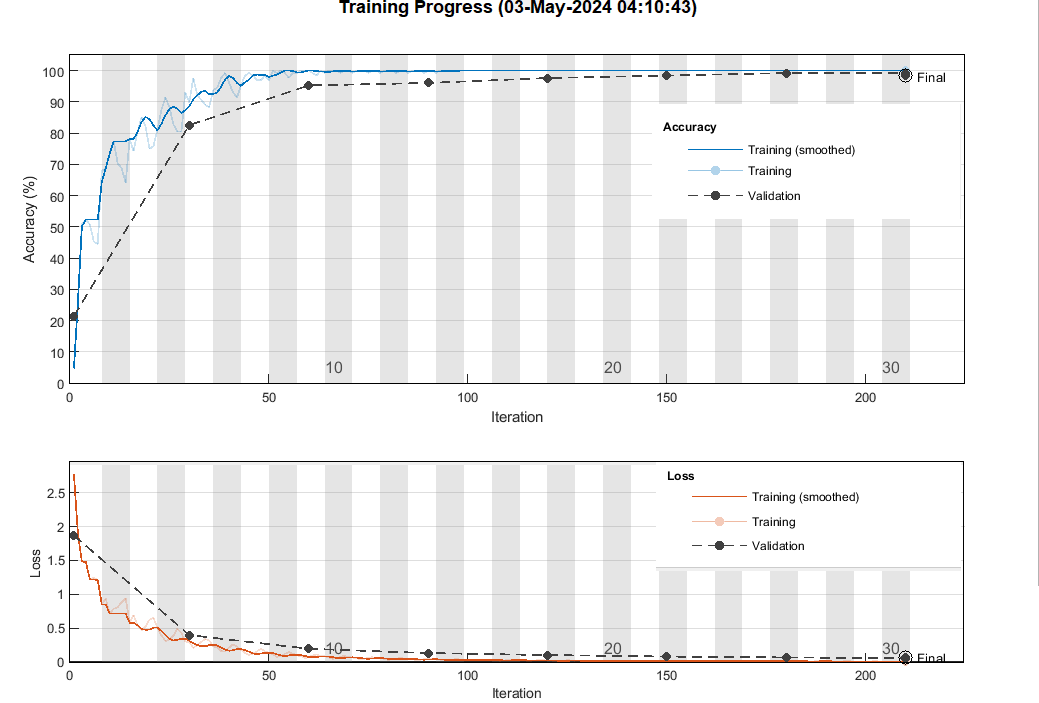


Figure ‎III‑10:Diagram of the 2d-DCNN model the training (Adam 32**×**32)

The training plots show the progress of a machine learning model during training. The top plot displays the accuracy curves over iterations for the training, smoothed training, and validation datasets. The validation accuracy reaches an impressive 99.58% by the final iteration.

The bottom plot shows the corresponding loss curves, which steadily decrease as training progresses, indicating the model is learning and minimizing the loss function effectively.

The model was trained for 30 epochs using a piecewise learning rate schedule with a learning rate of 0.0001. The total number of iterations was 1020.

These results are highly successful, with the model achieving near-perfect validation accuracy of over 99.5% while also minimizing the loss to very low levels. The plots suggest the training converged smoothly, and the model generalized exceptionally well to the validation data.

#### input image data a size (64\*64)

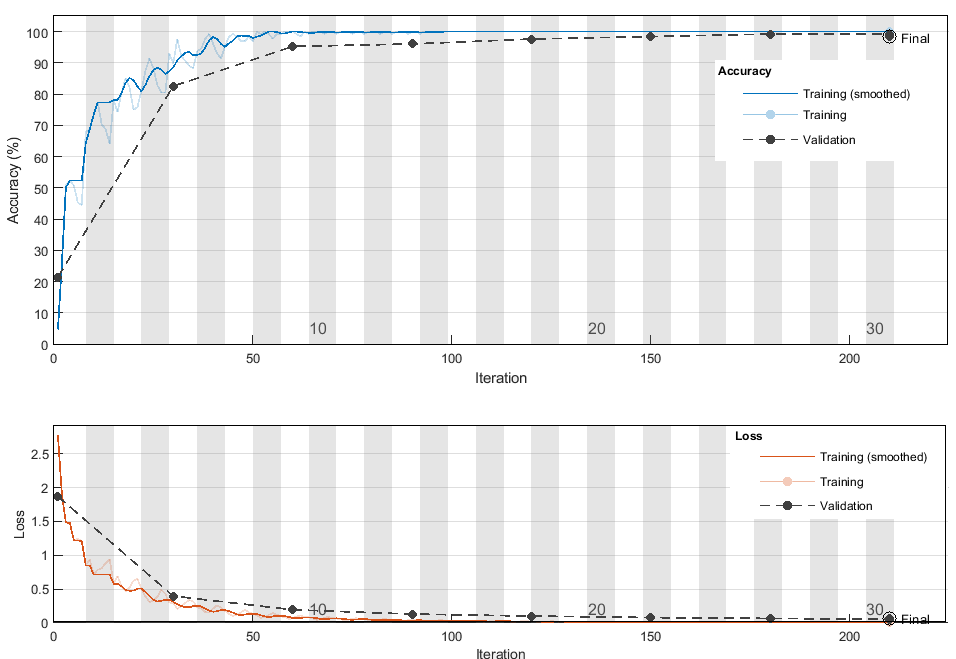
****

Figure ‎III‑11:Diagram of the 2d-DCNN model the training (Adam 64×64)

The training overview shows the model achieved a validation accuracy of 98.41% after 30 epochs, with a learning rate of 0.0001 using a piecewise schedule. Despite training for only 210 iterations, which is relatively low compared to the previous 1020 iterations, the model still managed to reach an impressive 98.41% validation accuracy. This high accuracy, close to the previous 99.58%, suggests the model architecture and training process were highly effective and data-efficient for the given task. The piecewise learning rate schedule likely helped the training converge faster while still achieving top performance. Although the number of iterations was modest, these results demonstrate successful model training, with the final validation accuracy attained within a reasonable computational budget. The model appears to generalize very well to unseen data, making it a high-performing solution for the problem at hand.

## SGD and Adam Results Comparaison:

* + 1. Case 01 :(32×32 pixel) input data size

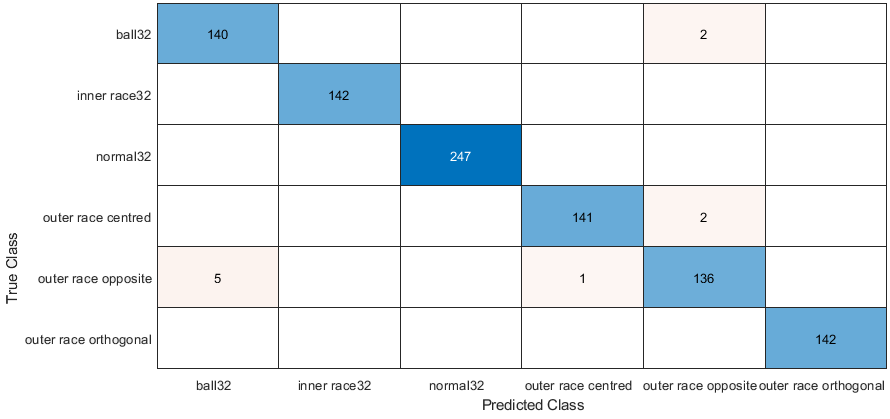


Figure ‎III‑12: confusion matrix of SGD optimizer (32×32)

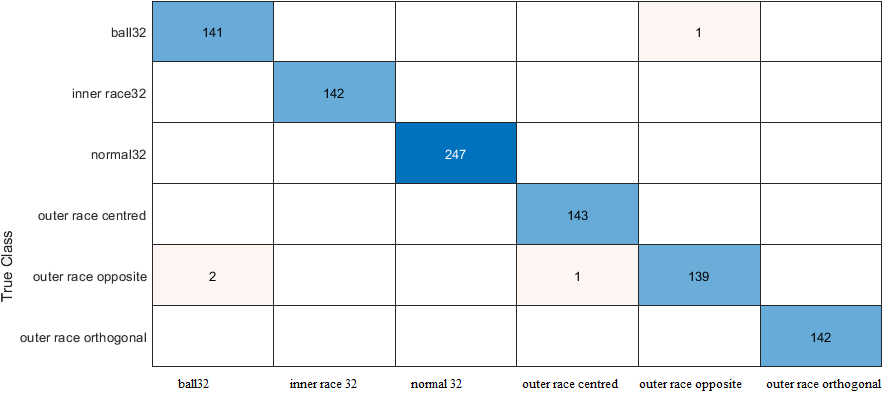


Figure ‎III‑13: confusion matrix of Adam optimizer (32×32)

**Table 6**:summarizing the classification result of SGD 32

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **True class** | **Predicted class** | **Correct** | **incorrect** | **precision** | **False positive rate** |
| **Ball 32** | **Ball 32** | 140 | - | 98.6% | 1.4% |
|  | **Other classes** |  | 2 |  |  |
| **Inner race32** | **Inner race32** | 142 | - | 100% | 0.0% |
| **Normal32** | **Normal32** | 247 | - | 100% | 0.0% |
| **Outer race centred** | **Outer race centred** | 141 | - | 98.6% | 1.4% |
|  | **Outer race opposite** | - | 2 |  |  |
| **Outer race opposite** | **Outer race opposite** | 136 | - | 95.8% | 4.2% |
|  | **Outer race centred** | - | 1 |  |  |
|  | **Other classes** | - | 5 |  |  |
| **Outer race orthogonal** | **Outer race orthogonal** | 142 | - | 100% | 0% |

**Table 7**:summarizing the classification result of Adam 32

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **True class** | **Predicted class** | **Correct** | **incorrect** | **precision** | **False positive rate** |
| **Ball 32** | **Ball 32** | 141 | - | 99.3% | 0.7% |
|  | **Other classes** |  | 1 |  |  |
| **Inner race32** | **Inner race32** | 142 | - | 100% | 0.0% |
| **Normal32** | **Normal32** | 247 | - | 100% | 0.0% |
| **Outer race centred** | **Outer race centred** | 143 | - | 100% | 0.0% |
| **Outer race opposite** | **Outer race opposite** | 139 | - | 97.9% | 2.1% |
|  | **Outer race centred** | - | 1 |  |  |
|  | **Other classes** | - | 2 |  |  |
| **Outer race orthogonal** | **Outer race orthogonal** | 142 | - | 100% | 0% |

#### Comparison:

* For the 'ball32' class, the Adam model has slightly better performance with 141 correct predictions (compared to 140), higher precision (99.3% vs. 98.6%), and lower false positive rate (0.7% vs. 1.4%).
* For the 'outer race centred' class, the Adam model has slightly better performance with 143 correct predictions (compared to 141), perfect precision (100% vs. 98.6%), and zero false positive rate (vs. 1.4%).
* For the 'outer race opposite' class, the Adam model has slightly better performance with 139 correct predictions (compared to 136), higher precision (97.9% vs. 95.8%), and lower false positive rate (2.1% vs. 4.2%).
* The number of misclassifications for the 'outer race opposite' class is slightly different, with 2 instances misclassified as 'outer race centred' in the second image/table (compared to 1 in the first), and 2 instances misclassified as other classes (compared to 5 in the first).

The classification performance appears to have slightly improved in the Adam algorithm with higher precision and lower false positive rate for some classes, particularly ‘ball32’’outer race centred’ and ‘outer race opposite’

* + 1. Case 02 :(64\*64 pixels) input data size

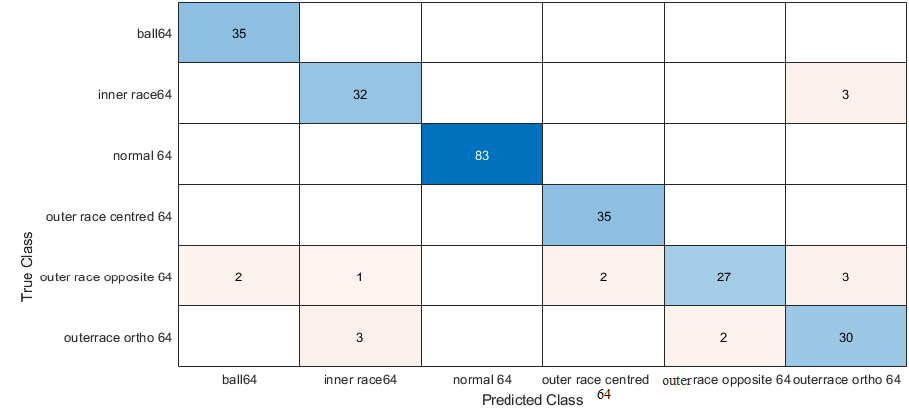


Figure ‎III‑14 confusion matrix of SGD optimizer (64×64)

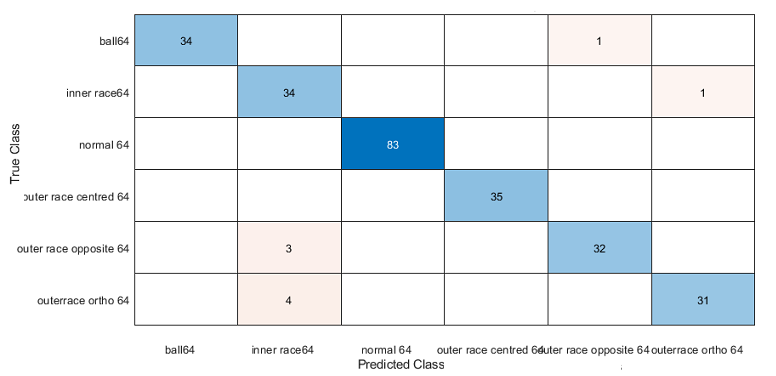


Figure ‎III‑15: confusion matrix of Adam (64×64) optimizer

**Table 8**:summarizing the classification result of SGD (64×64 pixel)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **True class** | **Predicted class** | **Correct** | **incorrect** | **precision** | **False positive rate** |
| **Ball 64** | **Ball 64** | 35 | - | 100% | 0.0% |
| **Inner race32** | **Inner race 64** | 142 | - | 100% | 0.0% |
|  | **Other classes** | - | 3 |  |  |
| **Normal32** | **Normal 64** | 83 | - | 100% | 0.0% |
| **Outer race centred** | **Outer race centred** | 35 | - | 100% | 0.0% |
| **Outer race opposite** | **Outer race opposite 64** | 27 | - | 77.1% | 22.9% |
|  | **Ball 64** |  | 2 |  |  |
|  | **Inner race 64** |  | 1 |  |  |
|  | **Outer race centred 64** | - | 2 |  |  |
|  | **Outer race ortho 64** |  | 3 |  |  |
| **Outer race orthogonal** | **Outer race ortho 64** | 30 | - | 85.7% | 14.3% |
|  | **Inner race 64** | - | 3 |  |  |

**Table 9**:summarizing the classification result of Adam 64

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **True class** | **Predicted class** | **Correct** | **incorrect** | **precision** | **False positive rate** |
| **Ball 32** | **Ball 64** | 34 | - | 97.1% | 2.9% |
|  | **Other classes** |  | 1 |  |  |
| **Inner race32** | **Inner race64** | 34 | - | 97.1% | 2.9% |
|  | **Other classes** | - | 1 |  |  |
| **Normal32** | **Normal64** | 83 | - | 100% | 0.0% |
| **Outer race centered** | **Outer race centered64** | 35 | - | 100% | 0.0% |
| **Outer race opposite** | **Outer race opposite 64** | 32 | - | 97.9% | 2.1% |
|  | **Other classes** | - | 3 |  |  |
| **Outer race orthogonal** | **Outer race orthogonal64** | 31 | - | 88.6% | 11.4% |
|  | **Other classes** | - | 4 |  |  |

#### Comparison:

* The same set of 6 classes is used for classification in both models: 'ball64', 'inner race64', 'normal64', 'outer race centred 64', 'outer race opposite 64', and 'outer race ortho 64'.
* The classes 'normal64' and 'outer race centred 64' have perfect precision (100%) and zero false positive rate in both models.
* For the 'ball64' class, the Adam model has slightly lower precision (97.1% vs. 100%) and higher false positive rate (2.9% vs. 0%).
* For the 'inner race64' class, the Adam model has higher precision (97.1% vs. 91.4%) and lower false positive rate (2.9% vs. 8.6%).
* For the 'outer race opposite 64' class, the Adam model has higher precision (91.4% vs. 77.1%) and lower false positive rate (8.6% vs. 22.9%).
* For the 'outer race ortho 64' class, the Adam model has slightly higher precision (88.6% vs. 85.7%) but also a higher false positive rate (11.4% vs. 14.3%).
* In the SGD model, there are more instances of misclassifications between certain classes (e.g., 'outer race opposite 64' misclassified as 'ball64', 'inner race64', 'outer race centred 64', and 'outer race ortho 64').

Overall, the classification performance appears to be slightly better in the Adam model for some classes ('inner race64', 'outer race opposite 64'), while the SGD model has better performance for the 'ball64' class. However, the differences are relatively minor, and both images exhibit generally good classification performance for most classes.

## Conclusion:

In conclusion, this work has demonstrated the efficacy of employing 2D Convolutional Neural Networks (CNNs) for the task of bearing fault detection and classification. By leveraging the powerful feature learning capabilities of CNNs and exploiting the spatial information inherent in the vibration signal data, the proposed approach has exhibited excellent performance in accurately identifying various bearing fault conditions. The comprehensive experimentation, including the evaluation of different input image sizes (32x32 and 64x64) and optimization algorithms (SGD and Adam), has provided valuable insights into the model's behavior and optimal configurations. The Adam optimizer, in particular, has proven to be a superior choice, facilitating faster convergence and better generalization performance. Additionally, the data augmentation technique employed has played a crucial role in mitigating overfitting and enhancing the model's robustness. The promising results obtained in this study highlight the potential of deep learning-based approaches for bearing fault diagnosis and open up avenues for further research in this domain, ultimately contributing to the advancement of predictive maintenance strategies in industrial applications.

**General Conclusion:**

The study presented in this document has demonstrated the effectiveness of employing deep learning techniques, specifically Convolutional Neural Networks (CNNs), for fault detection and classification in rotating electrical machines. By utilizing the powerful feature learning capabilities of CNNs and exploiting the spatial information inherent in vibration signal data, the proposed approach has exhibited excellent performance in accurately identifying various bearing fault conditions.

Through comprehensive experimentation and evaluation of different configurations, such as input image sizes and optimization algorithms, valuable insights have been gained regarding the model's behavior and optimal settings. The study highlights the superiority of the Adam optimization algorithm, which facilitated faster convergence and better generalization performance compared to the classical Stochastic Gradient Descent (SGD) algorithm.

Furthermore, the data augmentation technique employed played a crucial role in mitigating overfitting and enhancing the model's robustness, underscoring the importance of effective data preprocessing strategies in deep learning applications.

The promising results obtained in this research open up avenues for further exploration and advancement in deep learning-based approaches for predictive maintenance strategies in industrial applications. By leveraging the potential of CNNs and other deep learning architectures, it is possible to develop more efficient and reliable fault detection systems, contributing to increased productivity, reduced downtime, and enhanced overall operational efficiency.

As deep learning techniques continue to evolve, and computational resources become more accessible, the integration of these advanced algorithms into industrial fault detection and monitoring systems holds immense promise. Interdisciplinary collaborations between researchers, engineers, and domain experts will be essential to fully harness the power of deep learning for predictive maintenance and fault diagnosis, ultimately driving innovation and optimization in various sectors of the industry.

**References**

[1] KAZZAZ S.A.S.A., Singh G.K., « Experimental investigations on induction machine

condition monitoring and fault diagnosis using digital signal processing techniques”, Electric

Power Systems Research, vol. 65, p. 197-221, Elsevier, 2003.

[2] BIANCHI Bolognani S., Pré M.D., « Strategies for the fault-tolerant current control of a five-phase permanent-magnet motor”, IEEE Transaction on industry Applications, vol.43, no.4, p.

960-970, July- August 2007.

[3]What Are Roller Bearings? *| Types and Applications*. (2024, February 15). Emerson Bearing.

[4] Gritli, Y. (2014, March 11). Diagnosis and Fault detection in Electrical Machines and Drives based on Advanced Signal Processing Techniques. https://doi.org/10.6092/unibo/amsdottorato/6238

[5]. (2013) J. Trigeassou. Diagnostics of Electrical Machines. John Wiley & Sons (Trigeassou, 2013).

[6] GAUTHIER J., Hammouri. H., Othman S., “A simple observer for non-linear systems

applications to bioreactors”, IEEE Transactions on Automatic Control, vol. 37, no. 6, p. 875-

880, 1992.

[7] BONNETT A.H., YUNG C., “Increased efficiency versus increased reliability”, IEEE

industry Applications Magazine, p.1077-2618, January-February 2008.

[8] Oppermann, A. (2023, December 12). *What Is Deep Learning and How Does It Work?* Built

[9] Mishra, M. (2021, December 15). Convolutional Neural Networks, Explained - Towards Data Science. *Medium*.

[10] GeeksforGeeks. (2024b, March 14). *Introduction to Convolution Neural Network*. GeeksforGeeks.

[11] Smith, W.A.; Randall, R.B. Rolling element bearing diagnostics using the Case Western Reserve University data: A benchmark study. Mech. Syst. Signal Process. 2015, 64, 100–131.

Annexe: result of learning

Training on single CPU.sgdm 32

Initializing input data normalization.

|======================================================================================================================|

| Epoch | Iteration | Time Elapsed | Mini-batch | Validation | Mini-batch | Validation | Base Learning |

| | | (hh:mm:ss) | Accuracy | Accuracy | Loss | Loss | Rate |

|======================================================================================================================|

| 1 | 1 | 00:00:04 | 20.31% | 21.64% | 1.9298 | 1.9400 | 1.0000e-04 |

| 2 | 30 | 00:00:15 | 60.16% | 50.47% | 1.0373 | 1.2108 | 1.0000e-04 |

| 3 | 50 | 00:00:21 | 64.06% | | 0.9281 | | 1.0000e-04 |

| 3 | 60 | 00:00:25 | 56.25% | 64.69% | 0.9944 | 0.9615 | 1.0000e-04 |

| 4 | 90 | 00:00:36 | 75.00% | 75.55% | 0.7538 | 0.7725 | 1.0000e-04 |

| 5 | 100 | 00:00:39 | 84.38% | | 0.6351 | | 1.0000e-04 |

| 6 | 120 | 00:00:46 | 92.19% | 81.64% | 0.4639 | 0.5973 | 1.0000e-04 |

| 7 | 150 | 00:00:57 | 85.16% | 86.80% | 0.4727 | 0.4605 | 1.0000e-04 |

| 8 | 180 | 00:01:08 | 96.09% | 90.70% | 0.2863 | 0.3680 | 1.0000e-04 |

| 9 | 200 | 00:01:14 | 95.31% | | 0.3105 | | 1.0000e-04 |

| 10 | 210 | 00:01:19 | 97.66% | 93.05% | 0.2261 | 0.3008 | 1.0000e-04 |

| 11 | 240 | 00:01:29 | 97.66% | 94.38% | 0.1909 | 0.2545 | 1.0000e-04 |

| 11 | 250 | 00:01:32 | 96.88% | | 0.2155 | | 1.0000e-04 |

| 12 | 270 | 00:01:40 | 97.66% | 95.47% | 0.1796 | 0.2194 | 1.0000e-04 |

| 14 | 300 | 00:01:50 | 100.00% | 96.33% | 0.1158 | 0.1924 | 1.0000e-04 |

| 15 | 330 | 00:02:01 | 98.44% | 96.80% | 0.1084 | 0.1716 | 1.0000e-04 |

| 16 | 350 | 00:02:07 | 100.00% | | 0.0768 | | 1.0000e-04 |

| 16 | 360 | 00:02:11 | 100.00% | 97.03% | 0.0778 | 0.1549 | 1.0000e-04 |

| 17 | 390 | 00:02:21 | 98.44% | 97.34% | 0.1103 | 0.1409 | 1.0000e-04 |

| 18 | 400 | 00:02:24 | 98.44% | | 0.0819 | | 1.0000e-04 |

| 19 | 420 | 00:02:32 | 98.44% | 97.66% | 0.0954 | 0.1297 | 1.0000e-04 |

| 20 | 450 | 00:02:43 | 98.44% | 98.05% | 0.0886 | 0.1202 | 1.0000e-04 |

| 21 | 480 | 00:02:53 | 98.44% | 98.44% | 0.1100 | 0.1119 | 1.0000e-04 |

| 22 | 500 | 00:03:00 | 99.22% | | 0.0690 | | 1.0000e-04 |

| 23 | 510 | 00:03:04 | 100.00% | 98.44% | 0.0666 | 0.1050 | 1.0000e-04 |

| 24 | 540 | 00:03:14 | 96.09% | 98.52% | 0.0910 | 0.0991 | 1.0000e-04 |

| 24 | 550 | 00:03:18 | 100.00% | | 0.0429 | | 1.0000e-04 |

| 25 | 570 | 00:03:25 | 100.00% | 98.59% | 0.0608 | 0.0938 | 1.0000e-04 |

| 27 | 600 | 00:03:36 | 99.22% | 98.59% | 0.0533 | 0.0891 | 1.0000e-04 |

| 28 | 630 | 00:03:46 | 98.44% | 98.59% | 0.0466 | 0.0852 | 1.0000e-04 |

| 29 | 650 | 00:03:53 | 99.22% | | 0.0534 | | 1.0000e-04 |

| 29 | 660 | 00:03:57 | 99.22% | 98.75% | 0.0962 | 0.0816 | 1.0000e-04 |

| 30 | 690 | 00:04:08 | 100.00% | 98.83% | 0.0397 | 0.0783 | 1.0000e-04 |

|======================================================================================================================|

**Training on single CPU.sgdm 64**

**Initializing input data normalization.**

**|======================================================================================================================|**

**| Epoch | Iteration | Time Elapsed | Mini-batch | Validation | Mini-batch | Validation | Base Learning |**

**| | | (hh:mm:ss) | Accuracy | Accuracy | Loss | Loss | Rate |**

**|======================================================================================================================|**

**| 1 | 1 | 00:00:04 | 3.91% | 5.56% | 3.0333 | 2.9094 | 1.0000e-04 |**

**| 5 | 30 | 00:00:38 | 85.16% | 85.71% | 0.4303 | 0.4604 | 1.0000e-04 |**

**| 8 | 50 | 00:01:01 | 97.66% | | 0.1419 | | 1.0000e-04 |**

**| 9 | 60 | 00:01:13 | 97.66% | 91.27% | 0.1410 | 0.2567 | 1.0000e-04 |**

**| 13 | 90 | 00:01:48 | 99.22% | 92.06% | 0.0792 | 0.2122 | 1.0000e-04 |**

**| 15 | 100 | 00:02:00 | 100.00% | | 0.0665 | | 1.0000e-04 |**

**| 18 | 120 | 00:02:23 | 100.00% | 94.44% | 0.0447 | 0.1817 | 1.0000e-04 |**

**| 22 | 150 | 00:02:58 | 100.00% | 95.24% | 0.0350 | 0.1630 | 1.0000e-04 |**

**| 26 | 180 | 00:03:33 | 100.00% | 96.03% | 0.0259 | 0.1484 | 1.0000e-04 |**

**| 29 | 200 | 00:03:56 | 100.00% | | 0.0246 | | 1.0000e-04 |**

**| 30 | 210 | 00:04:09 | 100.00% | 97.62% | 0.0340 | 0.1366 | 1.0000e-04 |**

**|======================================================================================================================|**

**Training on single CPU. Adam 32**

**Initializing input data normalization.**

**|======================================================================================================================|**

**| Epoch | Iteration | Time Elapsed | Mini-batch | Validation | Mini-batch | Validation | Base Learning |**

**| | | (hh:mm:ss) | Accuracy | Accuracy | Loss | Loss | Rate |**

**|======================================================================================================================|**

**| 1 | 1 | 00:00:11 | 20.31% | 25.47% | 2.3826 | 2.2049 | 1.0000e-04 |**

**| 2 | 40 | 00:00:22 | 43.75% | 39.22% | 1.2545 | 1.2885 | 1.0000e-04 |**

**| 3 | 50 | 00:00:24 | 50.78% | | 1.1242 | | 1.0000e-04 |**

**| 4 | 80 | 00:00:29 | 62.50% | 53.67% | 0.9604 | 1.0846 | 1.0000e-04 |**

**| 5 | 100 | 00:00:31 | 69.53% | | 0.9291 | | 1.0000e-04 |**

**| 6 | 120 | 00:00:35 | 71.09% | 65.47% | 0.8586 | 0.9474 | 1.0000e-04 |**

**| 7 | 150 | 00:00:39 | 78.91% | | 0.8089 | | 1.0000e-04 |**

**| 7 | 160 | 00:00:40 | 71.09% | 70.23% | 0.7803 | 0.8292 | 1.0000e-04 |**

**| 9 | 200 | 00:00:46 | 83.59% | 77.81% | 0.7168 | 0.7192 | 1.0000e-04 |**

**| 11 | 240 | 00:00:52 | 91.41% | 84.30% | 0.5369 | 0.6240 | 1.0000e-04 |**

**| 11 | 250 | 00:00:54 | 84.38% | | 0.5897 | | 1.0000e-04 |**

**| 13 | 280 | 00:00:58 | 92.97% | 87.42% | 0.4507 | 0.5379 | 1.0000e-04 |**

**| 14 | 300 | 00:01:00 | 91.41% | | 0.4312 | | 1.0000e-04 |**

**| 14 | 320 | 00:01:03 | 93.75% | 89.38% | 0.4123 | 0.4652 | 1.0000e-04 |**

**| 16 | 350 | 00:01:07 | 94.53% | | 0.3632 | | 1.0000e-04 |**

**| 16 | 360 | 00:01:09 | 92.97% | 92.42% | 0.3778 | 0.3978 | 1.0000e-04 |**

**| 18 | 400 | 00:01:15 | 97.66% | 94.61% | 0.2784 | 0.3452 | 1.0000e-04 |**

**| 20 | 440 | 00:01:20 | 88.28% | 95.16% | 0.3478 | 0.2978 | 1.0000e-04 |**

**| 20 | 450 | 00:01:21 | 98.44% | | 0.2510 | | 1.0000e-04 |**

**| 21 | 480 | 00:01:25 | 95.31% | 96.63% | 0.2658 | 0.2608 | 1.0000e-04 |**

**| 22 | 500 | 00:01:28 | 99.22% | | 0.1997 | | 1.0000e-04 |**

**| 23 | 520 | 00:01:31 | 96.09% | 97.33% | 0.1825 | 0.2270 | 1.0000e-04 |**

**| 24 | 550 | 00:01:35 | 99.22% | 97.56 | 0.1838 | | 1.0000e-04 |**

**| 25 | 560 | 00:01:36 | 95.31% | 98.03% | 0.2108 | 0.2019 | 1.0000e-04 |**

**| 27 | 600 | 00:01:42 | 99.22% | 98.27% | 0.1568 | 0.1787 | 1.0000e-04 |**

**| 28 | 640 | 00:01:47 | 97.66% | 99.66% | 0.1251 | 0.1599 | 1.0000e-04 |**

**| 29 | 650 | 00:01:48 | 99.22% | | 0.1107 | | 1.0000e-04 |**

**| 30 | 680 | 00:01:52 | 99.22% | 99.83% | 0.1206 | 0.1432 | 1.0000e-04 |**

**| 30 | 690 | 00:01:54 | 99.22% | 97.97% | 0.1033 | 0.1401 | 1.0000e-04 |**

**|======================================================================================================================|**

**Training on single CPU. Adam 64**

**Initializing input data normalization.**

**|======================================================================================================================|**

**| Epoch | Iteration | Time Elapsed | Mini-batch | Validation | Mini-batch | Validation | Base Learning |**

**| | | (hh:mm:ss) | Accuracy | Accuracy | Loss | Loss | Rate |**

**|======================================================================================================================|**

**| 1 | 1 | 00:00:04 | 3.91% | 9.56% | 3.0333 | 2.9094 | 1.0000e-04 |**

**| 5 | 30 | 00:00:38 | 85.16% | 87.71% | 0.4303 | 0.4604 | 1.0000e-04 |**

**| 8 | 50 | 00:01:01 | 97.66% | | 0.1419 | | 1.0000e-04 |**

**| 9 | 60 | 00:01:13 | 97.66% | 93.27% | 0.1410 | 0.2567 | 1.0000e-04 |**

**| 13 | 90 | 00:01:48 | 99.22% | 94.06% | 0.0792 | 0.2122 | 1.0000e-04 |**

**| 15 | 100 | 00:02:00 | 100.00% | | 0.0665 | | 1.0000e-04 |**

**| 18 | 120 | 00:02:23 | 100.00% | 95.44% | 0.0447 | 0.1817 | 1.0000e-04 |**

**| 22 | 150 | 00:02:58 | 100.00% | 96.24% | 0.0350 | 0.1630 | 1.0000e-04 |**

**| 26 | 180 | 00:03:33 | 100.00% | 97.03% | 0.0259 | 0.1484 | 1.0000e-04 |**

**| 29 | 200 | 00:03:56 | 100.00% | | 0.0246 | | 1.0000e-04 |**

**| 30 | 210 | 00:04:09 | 100.00% |** 98.41**% | 0.0340 | 0.1366 | 1.0000e-04 |**

**|======================================================================================================================|**