College of Excellence, nirf 2022-6th Rank

Autonomous and Affiliated to Bhaarathiar University

Reaccredited with A++ grade by NAAC, An ISO 9001:2015 Certified Institution

Peelamedu, Coimbatore – 641004

IMAGE ANALYTICS ON AIR COMPRESSORS

Project work submitted to PSGR Krishnammal College for women in partial fulfillment of the requirements for the award of the degree of Master of Science in Data Analytics
Bharathiar University, Coimbatore-641046

Done by

KOUSHIKA RG

(21MDA021)

Guided by

Dr. T.A.ALBINAA, M.Sc., M.Phil., Ph.D.,

Assistant Professor, Department of Data Analytics(PG)

DEPARTMENT OF DATA ANALYTICS (PG)

PSGR KRISHNAMMAL COLLEGE FOR WOMEN

Peelamedu, Coimbatore – 641004

www.psgrkcw.ac.in

APRIL 2023

CERTIFICATE

CERTIFICATE

This is to certify that this project work entitled " **Explore Image Analytics for automated exception detection in Pre-Delivery Inspection Process**" submitted to PSGR Krishnammal College For Women, Coimbatore in partial fulfilment of the requirements for the award of Master of Science in Data Analytics is a record of original work done **KOUSHIKA RG(21MDA021)** during her period of study in the Department of Data Analytics(PG), PSGR Krishnammal College for Women, Coimbatore under my Supervision and guidance and the project work has not formed the basis for the award of any other Degree/ Diploma / Associateship / Fellowship or any similar title to any candidate of any University.

Forwarded by

Dr.T.A.ALBINAA, M.Sc. M.Phil. PhD,
Faculty Guide

Dr.(Mrs.)N.RADHA.,M.Sc.,M.Phil.,Ph.D.,
Head of the Department

Submitted for viva-voice Examination held on 05 - 04 - 2023

Internal examiner

5/4/23
External examiner

DECLARATION

DECLARATION

I hereby declare that this project work entitled **"Explore Image Analytics for automated exception detection in Pre-Delivery Inspection Process"** submitted to PSGR Krishnammal College For Women, Coimbatore for the award of the Degree of Master of Science in Data Analytics, is a record of original work done by **KOSHIKA RG(21MDA021)** under the supervision and guidance of **Dr.T.A.ALBINAA M.Sc., M.Phil.,Ph.D.** Department of Data Analytics(PG), PSGR Krishnammal College for Women, Coimbatore and that this project work has not formed the basis for the award of any other Degree / Diploma / Associate ship / Fellowship or any similar title to any candidate of any University.

KOUSHIKA RG
(21MDA021)

Endorsed by

Dr.T.A.ALBINAA. M.Sc., M.Phil., Ph.D.,
Faculty Guide

PLACE: Coimbatore
DATE: 05-04-2023

ELGI

01-APRIL-2023

TO WHOMSOEVER IT MAY CONCERN

This is to certify that **Ms. KOUSHIKA R G (21MDA021)** of Final year M.Sc., Data Analytics, student of PSGR KRISHNAMMAL COLLEGE FOR WOMEN, Coimbatore has been completed her project work entitled as **"EXPLORE IMAGE ANALYTICS FOR AUTOMATED EXCEPTION DETECTION IN PRE-DELIVERY INSPECTION PROCESS"** in our concern from 19 December 2022 to 31 March 2023.

For ELGI EQUIPMENTS LIMTED,

Dr. B. Soma Sundaram
Head Digital Transformation

CONTENT

CONTENT

ACKNOWLEDGEMENT

ACKNOWLEDGEMENT

"Success is to be measured not so much by the position that one has reached in life but as by the obstacle which he had overcome while trying to succeed".

I extend my thanks to **Dr. (Mrs.) R. NANDINI, Chairperson,** PSGR Krishnammal College for Women, and Coimbatore for her full support and for all the resources provided.

I express my whole hearted thanks to **Dr. (Mrs.) N. YESODHA DEVI M.Com, M.Phil., Ph.D., and Secretary** PSGR Krishnammal College for Women, and Coimbatore for having given me the opportunity to undertake this project work.

I extend my thanks to **Dr. (Mrs.) P.Meena, M.Sc., M.Phil., Ph.D., Principal,** PSGR Krishnammal College for Women, and Coimbatore for her full support and for all the resources provided.

I extend my sincere thanks to **Dr.(Mrs.) Latha Parameshwaren,M.C.A.,Ph.D.,** Director, Computational Sciences, PSGR Krishnammal College for women, Coimbatore, for her kind support and contribution to the successful completion of the project work.

I am extremely grateful to **Dr. (Mrs.) N.Radha, M.Sc., M.Phil., Ph.D., Head, Department of Data Analytics (PG),**PSGR Krishnammal College for Women, and Coimbatore for her sustained interest and advice that have contributed to a great extent to the completion of the project work.

I wish my indebtedness to my guide **Dr.T.A.Albinaa, M.Sc., M.Phil., Ph.D.,** Department of Data Analytics (PG), and PSGR Krishnammal College for Women, Coimbatore for her appropriate guidance, suggestions and support in the completion of this project work.

I extend my sincere thanks to **Dr. B. Somasundaram, SMISA, Msc(DS), PhD(hc)., ELGI Equipments Ltd.,** Head of Digital Transformation, Coimbatore, for his kind support and contribution to the successful completion of the project work.

My sincere thanks to all my staff of Department of Data Analytics (PG) for their timely support and encouragement.

Finally, I place on record my deep sense of gratitude to my beloved parents and to my friends for their timely support in completing this project work.

SYNOPSIS

SYNOPSIS

Image analytics is the process of analysing and interpreting images using various techniques, including computer vision, Machine Learning, and statistical methods. The goal of image analytics is to extract meaningful insights and information from digital images. Image classification is a type of image analytics .Image classification is a task in computer vision that involves assigning a label or category to an image based on its content. The goal of image classification is to automatically classify images into specific categories, such as animals, plants, objects, or scenes. Automation has transformed the manufacturing industry in recent years by increasing productivity, reducing errors, and improving safety. With the rise of digital technologies, automation has become more advanced, with the use of Robotics, Artificial Intelligence, and Machine Learning. Automation is being used in the manufacturing industry is through AI and ML. These technologies can be used to analyze data from sensors, cameras, and other sources to improve manufacturing processes. The project seeks to automate the process of checking whether all the required images of compressor are present in a given file before dispatch of the product using deep learning. With the ever-increasing amount of data being generated in different fields, there is a need for efficient ways to manage, organize and analyze the data. Image classification is one of the essential tasks in managing image data, and it involves grouping similar images into different categories. However, manually classifying large amounts of images can be tedious and time-consuming. Therefore, automating this process using deep learning can save time and improve accuracy. To achieve the project's objective, a huge dataset of images has been collected. This dataset is used to train a convolutional neural network (CNN) to recognize the different classes of images. CNN is a type of deep learning architecture that can learn to identify visual patterns in data by processing images in multiple layers.

INTRODUCTION

1. INTRODUCTION

Image analytics also referred to as image analysis or image processing, is a process of analysing digital images using computer algorithms. It involves the extraction of meaningful information from visual data by applying techniques such as computer vision, Machine Learning, and Deep Learning to interpret the visual content of an image. The analysis of images has become a crucial part of various industries, including healthcare, manufacturing, security, retail, and entertainment.

In manufacturing, image analytics can be used to monitor production processes and identify defects in products. It can analyse images of products during the manufacturing process to detect defects such as cracks, surface deformations, and colour variations. This helps companies ensure that their products meet quality standards and reduce the cost of recalls.

Image analytics can also be used in security and surveillance to analyse footage from cameras and identify suspicious activities or individuals. It can analyse images in real-time to detect objects, track movements, and recognize faces. This can help prevent crimes and improve public safety. In retail, image analytics can be used to analyse customer behaviour and preferences. It can analyse images from in-store cameras to understand how customers move around the store, which products they look at, and how long they stay in certain areas. This information can be used to optimize store layout, product placement, and marketing strategies to improve the customer experience and increase sales.

Overall, image analytics has numerous applications and can provide valuable insights by automating the analysis of visual data, reducing human error, and increasing efficiency. It is an important tool for decision-making and can help companies and organizations in various industries optimize their operations, improve product quality, and enhance customer experience.

Automation has transformed the manufacturing industry in recent years by increasing productivity, reducing errors, and improving safety. With the rise of digital technologies, automation has become more advanced, with the use of Robotics, Artificial Intelligence, and Machine Learning. Automation is being used in the manufacturing industry is through AI and ML. These technologies can be used to analyse data from sensors, cameras, and other sources to improve manufacturing processes. The importance of image analytics in real-time industries is significant. image analytics can be used for quality control, defect detection, and predictive maintenance. By analysing images of products, manufacturers can identify defects and quality issues early in the production process, reducing waste and improving product quality. Image analytics can also be used for predictive maintenance by analysing images of machinery and identifying potential issues before they cause downtime or failure. Overall, automating the process of checking whether all the classes of images of air compressors are present in a given file using deep learning is a promising solution that can save time and improve accuracy in managing image data. This project can benefit a wide range of industries that use air compressors, such as manufacturing, construction and automotive.

1.1 ORGANIZATION PROFILE

ELGi Equipments Limited is a global air compressor manufacturer with a broad line of innovative and technologically superior compressed air systems. With a presence across 120 countries worldwide and a portfolio of over 400 products, ELGi is committed to ensuring its customers achieve their productivity goals while keeping the cost of ownership low. At ELGi is believe in forging long-term partnerships with our stakeholders worldwide partnerships guided by reliability, technology, capability, and values, fuelled by innovation and a vision is to “Always be the choice everywhere”.

ELGi offers a complete range of compressed air solutions from oil lubricated and oil-free rotary screw compressors, oil lubricated and oil-free reciprocating compressors and centrifugal compressors, to dryers, filters and downstream accessories.

ELGi has recently introduced the world’s smallest oil-free screw air compressor, the AB 'Always Better' Series. A disruption in oil-free compressed air technology, this range offers customers a solution at approximately 8 - 10 per cent reduced life cycle cost compared with the prevailing oil-free technology. With significantly lower maintenance and ease of use, customers can be assured of reliable, high air purity for sensitive industrial applications.

1.2 PROBLEM STATEMENT

The problem statement for this project is to automate the process of quality analysis for an air compressor company using image classification. The company has a requirement to ensure that pictures of air compressors are taken and stored in the appropriate folders before the dispatch Currently, this process is manual and there is a possibility that some pictures may be missed. Therefore, the company wants to develop a system that can automatically detect and classify images of air compressors based on their category, such as make, model, and type. The proposed solution involves developing an image classification algorithm that can analyse images of air compressors and identify their category. The algorithm should be able to differentiate between different categories of images and ensure that all images of each category are captured and stored in the appropriate folder. The system should also be able to alert the quality analyst if any of the categories of images are missing, so that corrective action can be taken.

1.2 TOOL DESCRIPTION

JUPYTER NOTEBOOK IN VS CODE

Notebook is an open-source web applications that allows creating and sharing codes and documents. It provides an environment, the code can be documented, run it, look at the outcome, visualize data and see the results without leaving the environment. This makes it a handy tool for performing end to end data science workflows – data cleaning, statistical modelling, building and training machine learning models, visualizing data, and many, many other uses. Jupyter Notebooks really shine even in the prototyping phase. This is because the code is written in independent cells are executed individually. This allows the user to test a specific block of code in a project without having to execute the code from the start of the script. Many other IDE environments (like RStudio) also do these in several ways, but it is found that Jupyter's individual cells structure to be the best of the lot. These Notebooks are incredibly flexible, interactive and powerful tools in the hands of a data scientist. They even allow run other languages besides Python, like R, SQL, etc. Since they are more interactive than an IDE platform, they are widely used to display codes in a more pedagogical manner.

FEATURES OF JUPYTER NOTEBOOK:

Jupyter Notebook was created to make it easier to show one's programming work, and to let others join in. Jupyter Notebook allows you to combine code, comments, multimedia, and visualizations in an interactive document called a notebook, naturally that can be shared, re-used, and re-worked.

DATA VISUALIZATION:

Most people have their first exposure to Jupyter Notebook by way of a data visualization, a shared notebook that includes a rendering of some data set as a graphic. Jupyter Notebook lets you author visualizations, but also share them and allow interactive changes to the shared code and data set.

CODE SHARING:

Cloud services like GitHub and Pastebin provide ways to share code, but they're largely non-interactive. With a Jupyter Notebook, you can view code, execute it, and display the results directly in your web browser.

LIVE INTERACTIONS WITH CODE:

Jupiter Notebook code isn't static; it can be edited and re-run incrementally in real time, with feedback provided directly in the browser. Notebooks can also embed user controls (e.g., sliders or text input fields) that can be used as input sources for code

VISUAL STUDIO CODE

Visual Studio Code, also commonly referred to as VS Code, is a source-code editor made by Microsoft with the Electron Framework, for Windows, Linux and macOS. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git. Users can change the theme, keyboard shortcuts, preferences, and install extensions that add functionality.

In the Stack Overflow 2022 Developer Survey, Visual Studio Code was ranked the most popular developer environment tool among 71,010 respondents, with 74.48% reporting that they use it.

Features of VS Code

Visual Studio Code is a source-code editor that can be used with a variety of programming languages, including

- C
- C#,
- C++
- Fortran
- Go
- Java
- JavaScript
- Node.js
- Python
- Rust.

It is based on the Electron framework, which is used to develop Node.js web applications that run on the Blink layout engine. Visual Studio Code employs the same editor component (codenamed "Monaco") used in Azure DevOps (formerly called Visual Studio Online and Visual Studio Team Services).

PYTHON

Guido van Rossum, developed the python language. It is an easy to learn, powerful programming language. It has efficient high-level data structures and a simple but effective approach to object-oriented programming.

Python's elegant syntax and dynamic typing, together with its interpreted nature, make it an ideal language for scripting and rapid application development in many areas on most platforms.

Features of Python

Python is a simple and minimalistic language. Reading a good Python program feels almost like reading English, although very strict English! It allows you to concentrate on the solution to the problem rather than the language itself.

Easy to Learn

Python is extremely easy to get started with. Python has an extraordinarily simple syntax, as already mentioned.

Free and Open Source

Python is an example of a FLOSS (Free Library and Open Source Software). In simple terms, you can freely distribute copies of this software, read its source code, make changes to it, and use pieces of it in new free programs. FLOSS is based on the concept of a community which shares knowledge. This is one of the reasons why Python is so good to use and it is constantly improved for better usage.

High-level Language

Python is a high-level language since the programmer need to bother about the low- level details such as managing the memory etc,

Portable

Python has been ported to (i.e. changed to make it work on) many platforms. Python programs can work on any of these platforms without requiring any changes. Python can run on GNU/Linux, Windows, FreeBSD, Macintosh, Solaris, OS/2, Amiga, AROS, AS/400, BeOS, OS/390, z/OS, Palm OS, QNX, VMS, Psion, Acorn RISC OS, VxWorks, PlayStation, Sharp Zaurus, Windows CE and PocketPC

Interpreter

Python does not need compilation to binary. Python converts the source code into an intermediate form called byte codes and then translates this into the native language of our computer. All this, actually, makes using Python much easier since the programmer don't have to worry about compiling the program, making sure that the proper libraries are linked and loaded, etc.

Object Oriented (OOPS)

Python supports procedure-oriented programming as well as object-oriented programming. In procedure-oriented languages, the program is built around procedures or functions which are nothing but reusable pieces of programs. Python has a very powerful but simplistic way of doing OOP, especially when compared to big languages like C++ or Java.

Extensible

If someone can run piece of code to run very fast or want to have some piece of algorithm not to be open, then they can code that part of the program in C or C++ and then use it from their Python program.

Extensive Libraries

The Python Standard Library is huge indeed. It can help the programmer to do various things involving regular expressions, documentation generation, unit testing, threading, databases, web browsers, CGI, FTP, email, XML, XML-RPC, HTML, WAV files, cryptography, GUI (graphical user interfaces), and other system-dependent stuff.

APPLICATIONS OF PYTHON

GUI-Based Desktop Applications

Python has simple syntax, modular architecture, rich text processing tools and the ability to work on multiple operating systems which make it a desirable choice for developing desktop-based applications. There are various GUI toolkits like python, PyQt or PyGtk available which help developers create highly functional Graphical User Interface (GUI). The various applications developed using Python includes:

Image Processing and Graphic Design Applications

Python has been used to make 2D imaging software such as Inks cape, GIMP, Paint Shop Pro and Scribus. Further, 3D animation packages, like Blender, 3ds Max, Cinema 4D, Houdini, Light wave and Maya, also use Python in variable proportions.

Scientific and Computational Applications

The higher speeds, productivity and availability of tools, such as Scientific Python and Numeric Python, have resulted in Python becoming an integral part of applications involved in computation and processing of scientific data. 3D modelling software, such as Free CAD, and finite element method software, such as Abaqus, are coded in Python.

Games

Python has various modules, libraries and platforms that support development of games. For example, PySoy is a 3D game engine supporting Python 3, and PyGame provides functionality and a library for game development. There have been numerous games built using Python including Civilization-IV, Disney's Toontown Online, and Vega Strike.

Enterprise and Business Applications

With features that include special libraries, extensibility, scalability and easily readable syntax, Python is a suitable coding language for customizing larger applications. Reddit, which was originally written in Common Lips, was rewritten in Python in 2005.

Language Development

Python's design and module architecture has influenced development of numerous languages. Boo language uses an object model, syntax and indentation, similar to Python. Further, syntax of languages like Apple's Swift, Coffee Script, Cobra, and OCaml all share similarity with Python.

Operating Systems

Python is often an integral part of Linux distributions. For instance, Ubuntu's Ubiquity Installer, and Fedora's and Red Hat Enterprise Linux's Anaconda Installer are written in Python. Gentoo Linux makes use of Python for Portage, its package management system.

PACKAGES

Pandas

Pandas is a popular Python library for data analysis. It is not directly related to Machine Learning. As the dataset must be prepared before training. In this case, Pandas comes handy as it was developed specifically for data extraction and preparation. It provides high-level data structures and wide variety tools for data analysis. Here, data selection and data pre-processing are done using pandas.

NumPy

NumPy is a very popular python library for large multi-dimensional array and matrix processing, with the help of a large collection of high-level mathematical functions. It is very useful for fundamental scientific computations in Machine Learning. In this project, as regression involves some mathematical calculations using NumPy.

OpenCV

OpenCV is a huge open-source library for computer vision, machine learning, and image processing. OpenCV supports a wide variety of programming languages like Python, C++, Java, etc. It can process images and videos to identify objects, faces, or even the handwriting of a human. When it is integrated with various libraries, such as Numpy which is a highly optimized library for numerical operations, then the number of weapons increases in your Arsenal i.e whatever operations one can do in Numpy can be combined with OpenCV.

Sklearn

Scikit-learn (Sklearn) is the most useful and robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction via a consistence interface in Python. This library, which is largely written in Python, is built upon NumPy, SciPy and Matplotlib.

TensorFlow

TensorFlow is a free and open-source software library for machine learning and artificial intelligence. Its flexible architecture allows for the easy deployment of computation across a variety of platforms and from desktops to clusters of servers to mobile and edge devices. TensorFlow computations are expressed as stateful dataflow graphs. It can be used across a range of tasks but has a particular focus on training and inference of deep neural networks.

Keras

Keras is a high-level, deep learning API developed by Google for implementing neural networks. It is written in Python and is used to make the implementation of neural networks easy. It also supports multiple backend neural network computation. The final layers in a CNN are fully densely connected layers. In Keras, these layers are created using the Dense class. The Multilayer Perceptron (MLP) part in a CNN is created using multiple fully connected layers. In Keras, a fully connected layer is referred to as a Dense layer

DOMAIN

2. DOMAIN – MANUFACTURING INDUSTRY

The manufacturing industry is a sector of the economy that involves the production of goods using various raw materials, tools, and machinery. The manufacturing process involves several steps, including design, raw material selection, production, assembly, testing, and quality control. The sector includes a wide range of industries, including automotive, aerospace, electronics, food and beverage, and consumer goods. The manufacturing industry has played a crucial role in the growth and development of economies worldwide, creating jobs and driving innovation. It is a major contributor to the global economy. The manufacturing industry has undergone significant changes over the past few decades, driven by advancements in technology and globalization.

Automation and digitalization have transformed the manufacturing process, leading to increased efficiency, higher productivity, and improved product quality. The industry has also become more globalized, with companies outsourcing production to low-cost countries and establishing supply chain networks across borders. The new technological revolution and the new industrial revolution are gaining momentum. We believe that the new era of 'Internet plus artificial intelligence (AI)', characterized by ubiquitous networks, data-drivenness, shared services, cross-border integration, automatic intelligence, and mass innovation, is coming. The rapid development and fusion of new AI technologies with Internet technologies, new-generation information technologies, new energy technologies, materials technology, and biotechnology is an essential part of this new era, which in turn will enable the game-changing transformation of models, means, and ecosystems in terms of their application to the national economy, well-being, and national security.

The manufacturing industry is a cornerstone of national economy, people's livelihood, and national security. The deep fusion of manufacturing technology with information communication technology, intelligent technology, and product-related expertise in particular, is enabling a game-changing transformation in terms of manufacturing models, manufacturing approaches, and its ecosystems. Compressor manufacturers are companies that design, produce, and distribute various types of compressors used in industries such as manufacturing, construction, automotive, and aerospace. Compressors are devices that are used to compress air or gas and convert them into a higher-pressure state. There are many types of compressors

Each type has its own advantages and is suitable for different applications. Compressor manufacturers produce compressors that vary in size, power, and capacity to meet the needs of different industries and applications. Compressor manufacturers typically have a team of engineers and designers who develop and improve their products based on industry needs and technological advancements. They use advanced manufacturing processes and technologies to produce high-quality compressors that are efficient, reliable, and cost-effective.

ELGi manufactures a wide range of air compressors for today's industrial requirements. They offer an extensive range of air compressors and accessories from time-tested, robust workhorses to advanced, sophisticated compressed air solutions. Their products are designed and manufactured to offer high-quality, reliable, and energy-efficient compressed air solutions across industries from agriculture to construction, mining, manufacturing and so on. There are different air compressors based on their uses, such as oil-free air compressors, oil-lubricated air compressors, piston air compressors, and more. Air compressor manufacturing involves designing, building, and testing various types of air compressors for industrial, commercial, and consumer use. An air compressor is a device that converts power (usually from an electric motor, diesel engine, or gasoline engine) into potential energy stored in pressurized air. The manufacturing process involves several steps, including design and engineering, component fabrication, assembly, and testing. The design and engineering phase involves determining the specific requirements of the compressor, such as the pressure and flow rate needed, and designing the components to meet those requirements. Component fabrication involves producing the various parts of the compressor, such as the cylinders, valves, and pistons. These components are typically made from metal, such as steel or aluminum, and are precision-machined to ensure accuracy and consistency.

Quality analysts in the manufacturing industry can benefit significantly from the application of artificial intelligence (AI) technologies. AI can help improve the accuracy and efficiency of quality control procedures, as well as provide valuable insights into areas for improvement. One way AI can be used in quality analysis is through the use of machine vision systems. These systems use cameras and advanced algorithms to inspect and analyse products, identifying defects and deviations from quality standards. Machine vision systems can quickly and accurately identify defects that may be missed by human inspectors, increasing the overall efficiency and accuracy of quality control procedures.

DATA MODELLING

3. DATA MODELING

3.1 DATASET DESCRIPTION

Number of images:1650

Number of classes :9

Image size: min 400,max:750

Image format: jpg

Image content: Nameplate ,Panel plate, Motor nameplate, Air end nameplate, leaked sticker, performance sticker, motor rotation sticker,rear,front view of compressor

Nameplate

panel nameplate

front view

Air end nameplate

motor nameplate

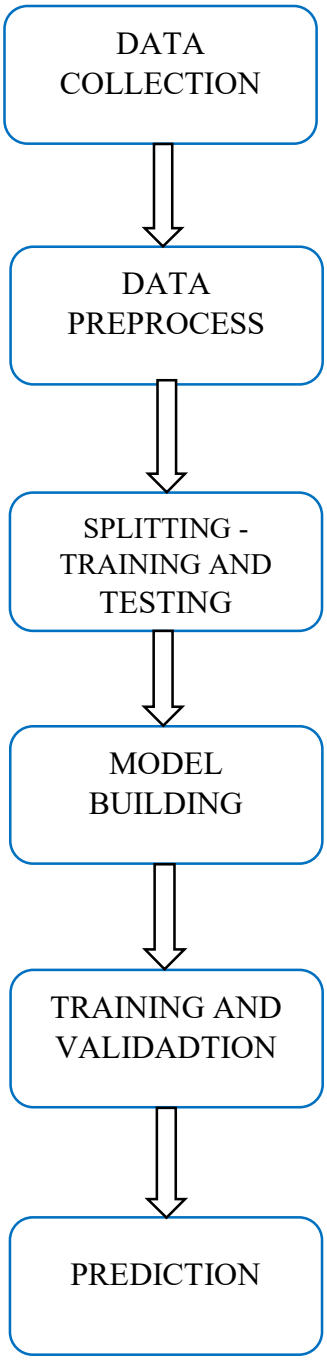
Rear view

Motor rotation sticker

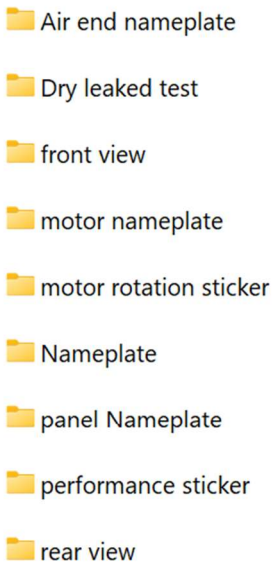
performance sticker

dry leaked test

3.2 PROCESS FLOW DIAGRAM



3.2.1 DATA COLLECTION



The dataset has been provided by the company, contains 1650 images,180 images in each class. The dataset is well-balanced and represents a range of images that the model will encounter in the real world. The dataset has been provided by the company

Images of an air compressor has been collected through photographs The photographs are used to capture visual information about the air compressor, such as its design, components, and condition.

Photographs can be an effective way to collect data because they provide a permanent and visual record of the air compressor, which can be easily shared and analysed.

To ensure the quality and accuracy of the data collected through photographs, it has been taken clear and high-quality pictures, using appropriate lighting and angles. The photographs should also be labeled and organized in a way that makes them easy to access and analyse. Additionally, obtained the necessary permissions and consents from any

individuals or organizations involved in the collection of the photographs, and to ensure that any ethical considerations are addressed

3.2.2 DATA PREPROCESSING

Pre-process the images, including resizing, cropping, and normalizing, to ensure that they are in the appropriate format for the model. Resizing: Images in a dataset may be of different sizes. Resizing involves scaling the images to a uniform size, which can help the model perform better.

Resizing an image to 300x300 pixels can be helpful for various reasons, such as reducing the size of the image for faster processing or standardizing the size of images for consistency in a dataset. However, resized and make sure that avoid result in a loss of quality or detail

```
img_width = 300
img_height = 300
EPOCHS = 30
BATCH_SIZE = 15
IMAGE_SHAPE = (img_width, img_height)
```

Cropping: Cropping involves removing unnecessary parts of the image that do not contain relevant information. This can help reduce noise and improve the model's performance. Cropping can also be used to standardize the aspect ratio of the images.

Normalizing: Normalizing involves scaling the pixel values in the image so that they fall within a specific range, typically between 0 and 1 or -1 and 1. Normalizing images can help improve the accuracy and efficiency of machine learning models by reducing the impact of different scales and ranges of pixel values across images. It can also help make images more visually comparable and interpretable.

Augmentation: Augmentation involves creating new images by applying transformations to the original images. This can help increase the size of the dataset and improve the model's performance. Transformations include rotating, flipping, and zooming the images.

Data augmentation can be implemented using various software or programming tools, such as Python and image processing libraries called OpenCV By applying different data

augmentation techniques to existing data, a larger and more diverse dataset can be created, which can help improve the accuracy and robustness of machine learning models.

```
datagen = ImageDataGenerator(rescale=1/255.,horizontal_flip=True,vertical_flip=True,height_shift_range=0.3,width_shift_range=0.5,brightness_range=[0.1,0.9],
rotation_range=90,validation_split=0.4,zoom_range=0.1)
train_gen = datagen.flow_from_directory(
    path,
    target_size=IMAGE_SHAPE,
    batch_size=BATCH_SIZE,
    class_mode='categorical',
    classes = dirs,
    subset='training',
    shuffle=True,
    seed=50,
)
```

However, it is important to be cautious when applied data augmentation techniques, as some techniques can introduce biases or distortions in the data. It is also important to avoid overfitting, where the model becomes too specialized to the training data and performs poorly on new data. To avoid overfitting, it has been used a combination of data augmentation techniques and regularization techniques, such as dropout or weight decay.

3.2.1 SPLITTING THE DATASET

split the data into a 80/20 train/test split using train_test_split(), and then further split the training data into a 80/20 train/validation split. The test_size argument specifies the percentage of the data that should be used for the testing set, the random_state argument ensures that the data is split in a reproducible way.

After splitting the data, we can use the resulting X_train, X_test, X_val, y_train, y_test, and y_val variables to train, validate, and test our model

MODEL BUILDING

4.MODEL BUILDING

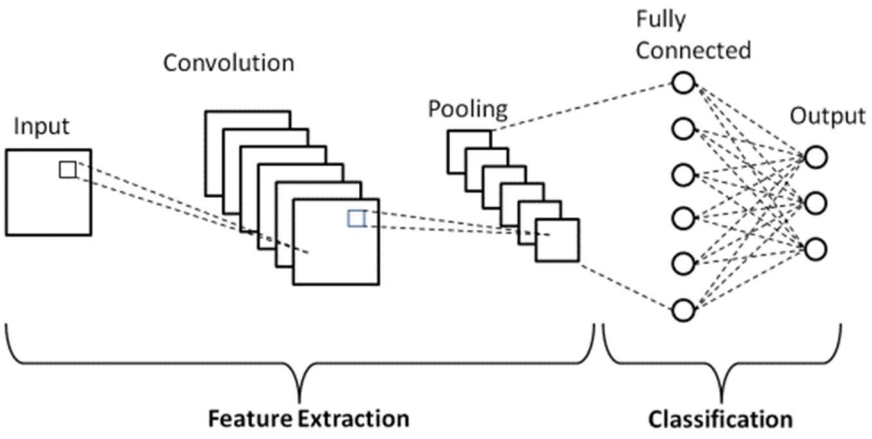
Convolutional Neural Network (CNN)

Convolutional neural networks (CNNs) are a type of deep learning model that are specifically designed for processing and analysing images and other multidimensional data. CNNs have revolutionized computer vision and have achieved state-of-the-art performance on a wide range of image-related tasks, including image classification, object detection, and semantic segmentation. At the heart of a CNN are convolutional layers, which apply a set of learnable filters to the input data. These filters are typically small in size (e.g., 3x3 or 5x5) but are applied across the entire input data in a sliding window fashion. Each filter is trained to detect a specific pattern or feature in the input data, such as edges, corners, or textures. The output of the convolutional layer is a set of feature maps that capture the presence of each filter's feature at various locations in the input.CNNs also often include other types of layers, such as pooling layers, which down sample the output of the convolutional layer by taking the maximum or average value of adjacent values, and fully connected layers, which combine the outputs of the previous layers to generate a final output. The combination of these layers allows CNNs to automatically learn and extract increasingly complex features from the input data, leading to highly accurate predictions or classifications. One of the key advantages of CNNs is their ability to learn hierarchical representations of the input data. As the input data is processed through the layers of the network, increasingly abstract and high-level features are extracted, with each layer building on the features learned by the previous layers. This hierarchical approach to feature extraction allows CNNs to capture both low-level features such as edges and corners, as well as higher-level features such as shapes and objects.

CNNs have been very successful in a wide range of applications, including image classification, object detection, and semantic segmentation. In image classification, CNNs are trained to assign an input image to one of several pre-defined classes. For example, a CNN could be trained to classify images of animals into categories such as cats, dogs, and birds. In object detection, CNNs are trained to detect and localize the presence of objects within an image. This can be useful for applications such as self-driving cars or surveillance systems. In semantic segmentation, CNNs are trained to classify each pixel in an image according to the object or background it belongs to, which can be useful for applications such as medical image analysis or satellite image interpretation.

Training CNNs requires a large amount of labeled data and significant computational resources, but there are many pre-trained models available that can be fine-tuned on specific tasks or domains. Transfer learning is a common approach in which a pre-trained CNN is used as a starting point, and then the final layers of the network are replaced or fine-tuned for the specific task at hand.

In summary, convolutional neural networks are a powerful and flexible type of deep learning model that have transformed the field of computer vision.



CONVOLUTION LAYERS

A typical convolutional neural network (CNN) consists of several types of layers, each of which performs a specific operation on the input data. The most common types of layers used in CNNs are:

Convolutional Layers: These layers apply a set of filters to the input data, with each filter detecting a specific feature or pattern in the input. The filters are typically small in size and are applied across the entire input data in a sliding window fashion. The output of the convolutional layer is a set of feature maps that capture the presence of each filter's feature at various locations in the input.

Activation Layers: These layers introduce nonlinearity into the network by applying an activation function to the

output of the previous layer. Common activation functions include Reu (Rectified Linear Unit), sigmoid, and tanh.

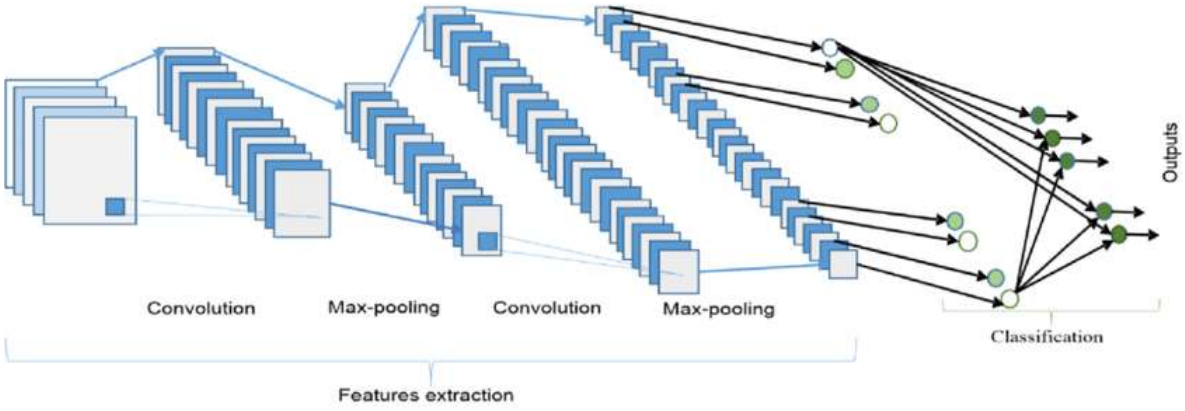
Pooling Layers: These layers down sample the output of the previous layer by taking the maximum or average value of adjacent values. Pooling layers help to reduce the size of the feature maps, which can help to speed up training and reduce overfitting.

Dropout Layers: These layers randomly drop out a fraction of the neurons in the previous layer during training. This helps to prevent overfitting by forcing the network to learn more robust features.

Batch Normalization Layers: These layers normalize the output of the previous layer by subtracting the mean and dividing by the standard deviation. Batch normalization can help to speed up training and improve the performance of the network.

Fully Connected Layers: These layers take the flattened output of the previous layer and compute a final output. Fully connected layers can be used for classification or regression tasks, depending on the nature of the problem.

4.1 Model Architecture



Choosing the layers for an image classification model can be a challenging task. When choosing the layers for your image classification model, it's important to consider the size and complexity of your dataset, as well as the available computational resources. In addition, you can experiment with different architectures and hyperparameters to find the best combination for your specific task. The layers that have been used in the model are base layer, global average pooling layer, dropout layer, dense layer

```python
model = Sequential()
base_model = InceptionV3(include_top=False,input_shape=(img_width, img_height,3))
base_model.trainable=False
model.add(base_model)
model.add(GlobalAveragePooling2D())
model.add(Dropout(0.50))
model.add(Dense(1024, activation='relu'))
model.add(Dense(len(dirs), activation='softmax'))
model.summary()
```

Python

Base layer:

he InceptionV3 model serves as the base layer in the CNN architecture and its weights are frozen during training. Freezing the pre-trained layers allows the model to leverage the learned features from the pre-trained model, which can improve performance while reducing required training time and computational resources. This is the base layer of the model, which uses the InceptionV3 architecture to extract features from the input images. The output shape of this layer is (8, 8, 2048), which means that it produces a feature map with a height of 8, a width of 8, and 2048 channels.

Global average pooling layer:

The global average pooling layer is a type of pooling layer commonly used in convolutional neural networks (CNNs) for image classification tasks. It reduces the dimensionality of the feature maps produced by the convolutional layers and computes the average value of each feature map, resulting in a single value for each feature. This technique reduces the number of parameters in the model and helps to improve its ability to generalize to new images by making it more robust to small changes in the input. Additionally, global average pooling can be an effective way to regularize the model and prevent overfitting, as it introduces a form of spatial averaging that can help to smooth out the features learned by the convolutional layers. This layer performs global average pooling on the output of the previous layer, which reduces the spatial dimensions of the feature map to 1x1 and computes the average value of each channel.

Dropout layer:

The dropout layer is a regularization technique commonly used in neural networks, including convolutional neural networks (CNNs) for image classification tasks. During

training, the dropout layer randomly drops out a certain percentage of the neurons in the previous layer, forcing the network to learn more robust and generalizable features. This technique can help to prevent overfitting, which occurs when the model becomes too specialized to the training data and fails to generalize to new data. By randomly dropping out neurons, the dropout layer forces the network to learn redundant representations of the data, making it more resistant to noise and increasing its ability to generalize to new images. This layer applies dropout regularization to the output of the previous layer, randomly setting a fraction of the values to zero during training to prevent overfitting.

Dense layer:

Dense layers, also known as fully connected layers, are a type of layer commonly used in neural networks, including convolutional neural networks (CNNs) for image classification tasks. Dense layers consist of a set of neurons that are fully connected to the previous layer, meaning that each neuron in the layer receives inputs from all the neurons in the previous layer. The main purpose of dense layers is to map the features extracted by the convolutional and pooling layers to the output classes. Each neuron in the dense layer computes a weighted sum of its inputs and applies a non-linear activation function, such as ReLU or sigmoid, to produce the output. The weights and biases of the neurons are learned during training using backpropagation, which adjusts the weights based on the error between the predicted outputs and the true labels. The number of neurons in the dense layers is a hyperparameter that can be tuned to optimize the performance of the model. In the case of the model you described, there are two dense layers with different numbers of units. The first dense layer has 1027 units, which is likely an arbitrary number chosen by the designer of the model. The second dense layer has a number of units equal to the total

number of classes in the dataset. This layer is used to output the predicted probabilities for each class.

This is a fully connected layer with 1024 units, which takes the output of the previous layer as input and performs a linear transformation. It has 2,098,176 trainable parameters.

This is the output layer of the model, which has 10 units (assuming a 9-class classification problem) and uses the softmax activation function to produce the final probability distribution over the classes. It has 5,125 trainable parameters.

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 inception_v3 (Functional)   (None, 8, 8, 2048)        21802784

 global_average_pooling2d_1   (None, 2048)            0
 (GlobalAveragePooling2D)

 dropout_1 (Dropout)         (None, 2048)              0

 dense_2 (Dense)             (None, 1024)              2098176

 dense_3 (Dense)             (None, 9)                 9125

=================================================================
Total params: 23,906,085
Trainable params: 2,103,301
Non-trainable params: 21,802,784
```

PREDICTION

5.PREDICTION

The implementation of an image classification algorithm to automate the quality analysis process for an air compressor company can yield significant results in terms of improving the quality of products, reducing errors, and increasing efficiency.

The first significant result is the accuracy and reliability of the quality analysis process. The manual process of quality analysis can be prone to human errors, such as missed images or misclassification of images, leading to inaccurate analysis and decision making. With the implementation of an image classification algorithm, the system can accurately classify images based on their category, such as make, model, and type. This ensures that all images are captured and stored correctly, reducing the likelihood of missing any images, and allowing for more accurate and reliable analysis of the data. As a result, the company can make better-informed decisions about the quality of their products and take corrective action if necessary, leading to improved customer satisfaction.

The second significant result is the efficiency and effectiveness of the quality analysis process. The manual process of quality analysis can be time-consuming and require a significant number of resources, leading to increased costs for the company. With the implementation of an image classification algorithm, the system can automate the process of image analysis, reducing the amount of time and resources required to complete the process. The algorithm can analyse images quickly and accurately, allowing the quality analyst to focus on other critical tasks that require human expertise. This results in increased efficiency, reduced costs, and faster for the analysis process.

Loaded Image


```
1/1 [==============================] - 5s 5s/step
Confidence:  80.53%
Class:  nameplate
```

```
1/1 [==============================] - 1s 601ms/step
Confidence:  99.74%
Class:  AE nameplate
```

Loaded Image

```
1/1 [==============================] - 0s 256ms/step
Confidence:  38.98%
Class:  rotation sticker
```

```
1/1 [==============================] - 0s 319ms/step
Confidence:  83.4%
Class:  panel nameplate
```

```
print(x1)
temp1 = []
for element in dirs:
    if element not in x1:
        temp1.append(element)

print(temp1,"not in the file")
```

```
[ leaked sticker, performance sticker, rear, front, motor nameplate] are not in folder
```

MODEL EVALUATION

6.MODEL EVALUATION

Evaluation and validation are crucial steps in image classification tasks to assess the performance of a machine learning model.

Evaluation involves measuring the accuracy of the model on a dataset that it has not seen before. This is typically done by splitting the dataset into two parts: a training set and a test set. The model is trained on the training set, and its performance is evaluated on the test set. The accuracy of the model on the test set gives an estimate of how well the model will perform on the new, unseen data.

Validation is the process of tuning the hyperparameters of the model to optimize its performance. Hyperparameters are parameters that are set before training, such as the learning rate or the number of layers in the neural network. Validation involves testing different combinations of hyperparameters on a validation set, which is a subset of the training data that is held out during training. The hyperparameters that give the best performance on the validation set are selected and the model is then trained on the full training set with those hyperparameters.

```
[ ]  1  test_loss, test_acc = model.evaluate(validation_gen)
     2  print('test accuracy : ', test_acc)
     3  trained_model =load_model('img_model.trained.weights.hdf5')
```

```
3/3 [==============================] - 3s 998ms/step - loss: 0.4666 - accuracy: 0.8421
test accuracy :  0.8421052694320679
```

Based on the provided output, the training accuracy is 91.7%, while the validation accuracy is 90%. This is a good sign that the model is generalizing well to new data, although the accuracy could potentially be improved.

The training loss is quite low at 0.320, indicating that the model is fitting well to the training data. The validation loss is slightly higher at 0.410 but still within a reasonable range.

Overall, the model appears to be performing well but could potentially benefit from some further optimization to improve the validation accuracy, such as trying different model architectures, tuning hyperparameters, or using data augmentation.

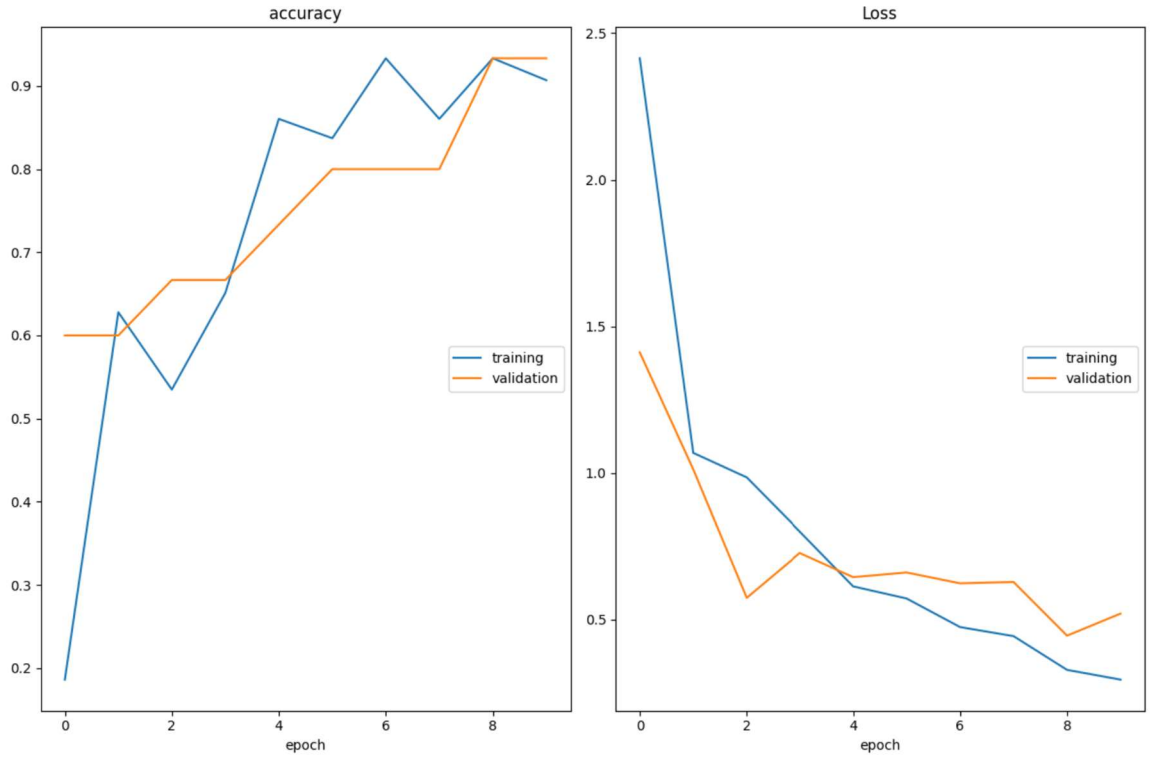


Fig:6.1 plot of accuracy and loss

```
accuracy
        training         (min:    0.500, max:    0.917, cur:    0.917)
        validation       (min:    0.600, max:    0.900, cur:    0.900)
Loss
        training         (min:    0.320, max:    1.060, cur:    0.320)
        validation       (min:    0.410, max:    0.900, cur:    0.410)
```

```
30/30 [==============================]-- 4s 1s/step - loss: 0.3195 -
accuracy: 0.9167 - val_loss: 0.4101 - val_accuracy: 0.9000
```

CONCLUSION

7.CONCLUSION

In conclusion, the development of an image classification algorithm to automate the quality analysis process for an air compressor company is a crucial step towards improving the quality of products and reducing errors in the production process. With the implementation of this solution, the company can save time and resources, improve the quality of their products, and increase customer satisfaction. The manual process of quality analysis can be prone to human errors, and there is a possibility that some pictures may be missed or misclassified, leading to a lack of information and incorrect analysis. By automating this process, the image classification algorithm can ensure that all images are captured and stored correctly, reducing the likelihood of missing any images, and allowing for more accurate and reliable analysis of the data. The system can alert the quality analyst if any categories of images are missing, making corrective action possible in a timely manner. This can prevent further errors from occurring and improve the overall quality of the products. Another significant advantage of the image classification algorithm is that it can differentiate between different categories of images, such as make, model, and type. This means that each image can be classified and stored in the appropriate folder, making it easier for the company to retrieve specific images when needed. The development of an image classification algorithm for an air compressor company is a crucial step towards improving the quality of products, reducing errors, and increasing customer satisfaction. The system can ensure that all images are captured and stored correctly it provide accurate and reliable analysis of the data, and alert the quality analyst if any categories of images are missing. By implementing this solution, the company can save time and resources, improve the quality of their products and stay competitive in the market.

BIBLIOGRPHY

8.BIBLIOGRAPHY

[1] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. "Gradient-based learning applied to document recognition." Proceedings

[2] Alexander Krull, Tim-Oliver Buchholz, and Florian Jug. "Noise2void-learning denoising from single noisy images." In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2129-2137. IEEE, 2019.

[3] Sanchez, F. Perronnin, T. Mensink, and J. Verbeek. "Image classification with the fisher vector: Theory and practice." International journal of computer vision 105, no. 3 (222-245).

[4] Arya, S., Mount, D.M., Netanyahu, N.S., Silverman, R., and Wu, A.Y. 1998. An Optimal algorithm for approximate nearest neighbour searching. Journal of the ACM, 45:891–923.

[5] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. "Reading digits in natural images with unsupervised feature learning." In NIPS Workshop on Deep Learning and Unsupervised Feature Learning, vol. 2011, pp. 1-9. 2011

[6] Lin, T. Y., Dollar, P. Girshick, R., He, K., Hariharan, B., & Belongie, S. "Feature Pyramid Networks for Object Detection." In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2117-2125, 2017.

[7] Kamavisdar, P., Saluja, S., & Agrawal, S. (2013). A survey on image classification approaches and techniques. International Journal of Advanced Research in Computer and Communication Engineering, 2(1), 1005–1009

[8] Y Bengio, A Courville and P. Vincent, "Representation learning: A review and new perspectives[J]", Pattern Analysis and Machine Intelligence IEEE Transactions on, vol. 35, no. 8, pp. 1798-1828, 2013.

[9] Shorten, C. and Khoshogoftaar, T. M., "A survey on Image Data, Augmentation for Deep Learning," Journal of Big Data Vol. 6 No. 60, pp. 1-48, 2019.

[10] Reiner U. Automatic analysis of Dewey decimal classification notations. In: Data analysis, machine learning and applications. 2008, pp. 697–704.

[11] Wang Binghao FanYaQin and Wei wang, "Depth study domestic research review [J]", China distance education, no. 6, pp. 2015-27.

[12] Jinfeng Liu, "A concise and effective to accelerate the Convolutional of the neural network method [J]", Science technology and engineering, vol. 33, pp. 240-244, 2014.

APPENDICES

APPENDIX

Libraries and packages

```
import tensorflow

from tensorflow import keras

from keras.preprocessing import image

from keras.preprocessing.image import ImageDataGenerator

from keras.models import Sequential, load_model

from keras.layers import Activation, Conv2D, Dense, Dropout, GlobalAveragePooling2D

from keras.regularizers import l1_l2

from keras.optimizers import Adam

from keras.callbacks import ModelCheckpoint, EarlyStopping

from keras.applications import InceptionV3

from sklearn.metrics import accuracy_score, precision_recall_fscore_support

from livelossplot import PlotLossesKeras

import keras.utils as image

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import warnings

warnings.filterwarnings('ignore')
```

```
import os
```

READ THE DIRECTORES

```
path="D:\data"
dirs=os.listdir(path)
for file in dirs:
    print(file)
```

IMAGE RESIZE

```
img_width = 300
img_height = 300
EPOCHS = 10
BATCH_SIZE = 15
IMAGE_SHAPE = (img_width, img_height)
```

DATASET SPLITTING

```
datagen=
ImageDataGenerator(rescale=1/255.,horizontal_flip=True,vertical_flip=True,height_shift_ra
nge=0.3,width_shift_range=0.5,brightness_range=[0.1,0.9],
rotation_range=90,validation_split=0.4,zoom_range=0.1)
train_gen = datagen.flow_from_directory(path,
    target_size=IMAGE_SHAPE,
```

```
    batch_size=BATCH_SIZE,
    class_mode='categorical',
    classes = dirs,
    subset='training',
    shuffle=True,
    seed=50,
)
validation_gen = datagen.flow_from_directory(
    path,
    target_size=IMAGE_SHAPE,
    batch_size=BATCH_SIZE,
    class_mode='categorical',
    classes = dirs,
    subset='validation',
    shuffle=True,
    seed=50,
)
```

MODEL BUILDING

```
model = Sequential()

base_model = InceptionV3(include_top=False,input_shape=(img_width, img_height,3))

base_model.trainable=False

model.add(base_model)

model.add(GlobalAveragePooling2D())

model.add(Dropout(0.50))

model.add(Dense(1024, activation='relu'))

model.add(Dense(len(dirs), activation='softmax'))

model.summary()
```

MODEL FITTING

```
steps_for_each_epoch = train_gen.samples

validation_steps_for_each_epoch = validation_gen.samples

OPTIMIZER = Adam(learning_rate=0.0001)

model.compile(loss='categorical_crossentropy',optimizer=OPTIMIZER,
metrics=['accuracy'])

checkpointer = ModelCheckpoint(filepath='img_model.trained.weights.hdf5', verbose=1,
save_best_only=True)
```

```
early_stop = EarlyStopping(monitor='val_loss', patience=10, restore_best_weights=True, mode='min')

history=model.fit(train_gen,epochs=EPOCHS,
steps_per_epoch=steps_for_each_epoch,validation_data=validation_gen,validation_steps=validation_steps_for_each_epoch,callbacks=[early_stop,checkpointer,PlotLossesKeras()],

verboe=True)
```

VALIDATION:

```
test_loss, test_acc = model.evaluate(validation_gen)

print('test accuracy : ', test_acc)

trained_model =load_model('img_model.trained.weights.hdf5')
```

PREDICTION:

```
x1=[]

def predict_img(img_path):

    img = image.load_img(path+img_path, target_size=(300, 300))

    img = image.img_to_array(img, dtype=np.uint8)

    img = np.array(img)/255.0

   plt.title("Loaded Image")

    plt.axis('off')

    plt.imshow(img.squeeze())

    plt.show()
```

```
    p = trained_model.predict(img[np.newaxis,...])
    print('Confidence: ',str(round(np.max(p[0])*100, 2))+'%')
    print('Class: ',dirs[np.argmax(p[0])])
    x1.append(dirs[np.argmax(p[0])])
    print("\n")
```

 # import the modules

```
import os
from os import listdir
```

 # get the path/directory

```
folder_dir = "D:/data/test"
for images in os.listdir(folder_dir):
```

 # check if the image ends with jpg

```
 if (images.endswith(".jpg")):
        x='/test/'+images
        predict_img(x)
```

#checking what are not in the file

```
print(x1)
dirs1=['AE nameplate', 'main namplate', 'motor nameplate', 'motor rotation sticker', 'panel
nameplate' leaked sticker, performance sticker, front view of compressor,rear view of
compressor]
```

```
temp1 = []

for element in dirs:

if element not in x1:

 temp1.append(element)

 print(temp1,"not in the file")
```