

EFFICIENT QUALITY ASSESSMENT OF BRICKS USING CONVOLUTIONAL NEURAL NETWORKS

Submitted in partial fulfillment of the requirements for the award of Bachelor
Engineering Degree in Electronics and Communication Engineering

By

DHARUN G (39130114)

DHEWESWAR B (39130116)



**DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
SCHOOL OF ELECTRICAL AND ELECTRONICS**

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with Grade “A” by NAAC|12B Status by UGC | Approved by AICTE

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119

APRIL-2023



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the Bonafide work of **G. DHARUN (Reg. no. 39130114)** and **B. DHEWESWAR (Reg. no. 39130116)** who have done the project work as a team who carried out the project entitled "**Efficient Quality assessment of Bricks using Convolutional Neural Networks**" under my supervision from October 2022 to April 2023.

Internal Guide

Mr. V. KALIST M.E

Head of the Department

Dr. T. RAVI, ME., Ph.D.

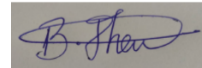
Submitted for Viva voce Examination held on 24.04.2023

INTERNAL EXAMINER

EXTERNAL EXAMINER

DECLARATION

We **DHARUN G (39130114)** and **DHEWESWAR B (39130116)** hereby declare that Project Report entitled “**EFFICIENT QUALITY ASSESSMENT OF BRICK USING CONVOLUTIONAL NEURAL NETWORKS**” done by us under the guidance of **Mr. KALIST. M.E** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Electronics and Communication Engineering.



DATE: 24/04/2023

SIGNATURE OF CANDIDATES

PLACE: CHENNAI

1. DHARUN G

2. DHEWESWAR B

ACKNOWLEDGEMENT

We are pleased to acknowledge our sincere thanks to Board of Management of SATHYABAMA for their kind encouragement in doing this project and for completing it successfully. We are grateful to them.

I convey my thanks to **Dr. N. M. NANDHITHA. M.E., Ph.D., Dean, School of Electrical and Electronics** and **Dr. T. RAVI. M.E., Ph.D., Head of the Department, Department of Electronics and Communication Engineering** for providing me necessary support and details at the right time during the progressive reviews.

We would like to express my sincere and deep sense of gratitude to our Project supervisor, **Mr. V. KALIST M.E** and our panel members for his valuable guidance, suggestions, and constant encouragement paved way for the successful completion of our project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the Department of Electronics and Communication who were helpful in many ways for the completion of the project.

ABSTRACT

Deep learning is a field of artificial intelligence which can be utilized in image recognition applications. By means of machine learning, software capable of solving complex tasks can be developed. In this work, a convolutional neural network was used as the machine learning method. The feasibility of using convolutional neural networks in masonry brick classification was studied in this Master's thesis. The aim was to discover the most efficient network model by testing different network structures. According to the preliminary development plan, the classification of brick types can be done in the brick factory. This is conducted by imaging bricks with a machine vision camera while the bricks are moving on a conveyor belt. After the imaging, a machine learning algorithm identifies different brick classes from the images. The bricks are sorted into different classes at the end of the production line, using conveyor belts and industrial robots. The content of the Master's thesis can be divided into three sections: the theoretical part, the image acquisition part and the results achieved with the network model. At first, topics related to the camera model, image formation and optics are covered. This is followed by the theory part, regarding machine learning and neural networks. Also, the network model used is presented. After the theoretical part, the equipment and the methods used in image acquisition are introduced. Finally, the results obtained with the network model are presented, and the suitability of the model for brick classification is evaluated. The bricks were pre-divided into three classes at the brick factory, based on different quality requirements. The imaging of the bricks was done in a laboratory, where 1105 bricks were imaged three times. A machine vision camera and lighting system were assembled on a conveyor belt. By using this system, each brick was imaged from the top, bottom and side surfaces. The machine vision camera used was controlled by vision software. With this equipment, the images were captured automatically as the

bricks moved along the conveyor belt. After the image acquisition the image data was modified into a suitable format for the machine learning algorithm. Two image datasets were used to test the convolutional neural network. The original dataset included 1800x1500 pixel size images. The second dataset included the same images. Additionally, in the second dataset images, the excessive background areas around the bricks were cropped. Both datasets contained 1105 images, consisting of the three brick classes. A pretrained VGG16 convolutional neural network was chosen as the network model. A transfer learning method was utilized with this network model. In transfer learning, pretrained network parameter values are used. Pretrained parameters were obtained for the VGG16 by training the network with the ImageNet database. Generally, network performance in a classification task can be improved by using transfer learning. This is especially the case when the number of images to be used in a classification task is limited. According to the results obtained using two different image size datasets, it can be concluded that the convolutional neural network is well suited for the classification of brick images. When performing image acquisition in factory conditions, it is important to keep the ambient lighting conditions constant during shooting. Acquiring a high-quality image dataset is an important thing in project.

CHAPTER .NO	TITLE	PAGE.NO
	ABSTRACT	v
	LIST OF FIGURES	x
	LIST OF TABLES	xi
	LIST OF ABBREVIATIONS	xii
1	INTRODUCTION	1
	1.1 Existing system	2
	1.1.1 Disadvantages	2
	1.2 Proposed System	3
	1.2.1 Advantages	3
	1.3 Difference b/w existing and proposed system	4
2	LITERATURE SURVEY	5
3	AIM AND SCOPE OF THE PROJECT	11
	3.1 Aim and Objective	11
	3.2 Problem Definition	12
	3.3 Problem Statement	13
4	METHODOLOGY	15
	4.1 INTRODUCTION	15
	4.1.1 What is CNN	15

4.2	Image Processing using Deep learning	15
4.3	Neural Networks	17
4.4	Basic neural network	18
4.4.1	Input layer	20
4.4.2	hidden layer	20
4.4.3	Output layer	21
4.5	types of neural networks	21
4.5.1	Convolutional Neural	21
4.5.2	Pooling operation	23
4.5.3	Fully connected layers	24
4.6	CNN Architecture	25
5	Software Required	26
5.1	python	26
5.1.1	Syntax	26
5.1.2	Libraries	26
5.1.3	Object Oriented Programming	26
5.1.4	Error Handling	26
5.1.5	Input Output(I/O)	27
5.1.6	Code Organization	27
5.1.7	Testing and debugging	27
5.1.8	Documentation	27
5.1.9	Community Resources	27

5.1.10	Conclusion	28
5.2	image processing in python	29
5.3	Algorithms tools and methods	29
5.4	python image processing	32
5.4.1	code 01	34
5.4.2	code 02	35
5.4.3	Dashboard	36
5.4.4	confusion matrices	37
5.4.5	Accuracy checking	38
6	Results and Discussion	39
6.1	Results	39
6.2	Discussion	39
7	Conclusion	
7.1	Conclusion	
7.2	Future Scope	41
7.3	References	44

LIST OF FIGURES

FIGURE	TITLE	PAGE NO
1.1	Basic neural network	17
1.2	CNN Architecture	25
3.1	CNN process	30
4.1	RGB IMAGE	30
4.2	python code 01	34
4.3	python code 02	35
4.4	Dashboard	36
4.5	Confusion Matrix	37
4.6	Accuracy Checking	38

LIST OF TABLES

Figure no	table	page
1.3	Difference between existed system Proposed system	4

LIST OF ABBREVIATIONS

ABBREVIATIONS

EXPANDS

CNN	CONVOLUTIONAL NEURAL NETWORK
POOL	POOLING LAYER
FCL	FULLY COONECTED LAYER
OCR	OPTICAL CHARACTER RECOGNITION
NLP	NATURAL LANGUAGE PROCESSING
RNN	RECURRENT NEURAL NETWORKS
DNN	DEEP NEURAL NETWORKS
LSTM	LONG SHORT-TERM MEMORY
DBN	DEEP BELIEF NETWORK
GAN	GENERATIVE ADVERISAL NETWORK
GRU	GATED RECURRENT UNITS
DRP	DIGITAL ROCK PHYSICS

CHAPTER 1

INTRODUCTION

Bricks belong to the oldest building elements in the world. The earliest findings of bricks can be dated to 7000 BC in southern Turkey, near the city of Jericho. Ancient bricks were made by hardening mud bricks in sunlight, in the warm regions of the world. The usage of fired bricks started in about 3500 BC, which also made it possible to manufacture bricks in cooler climate regions. The objective of this thesis was to examine the quality of masonry bricks, with an appropriate machine learning model. The topic of this thesis was suggested by a local engineering company, which received a commission from a brick factory to improve the defect detection of masonry bricks. The plan was to implement quality inspection from high quality images, captured by a machine vision system. The research questions were firstly to examine whether a quality inspection can be conducted using machine learning. Secondly, if a feasible model was found, the challenge was to create a useful application. Also, guidelines for appropriate imaging setup were to be planned, to be able to collect images for inspection. In factory facilities, masonry bricks are inspected manually by an operator and then divided into three different categories. There is growing interest in upgrading the automation level in the brick factory to enable more efficient production. In the factory in question, there are over 10 brick color options to choose from. Additionally, the brick size varies, and the outer surface can have five different patterns. One brick type was chosen for examination. The plan was to train the learning model using one brick type and then, if successful, it would be easy to train other types with the same model. The early plan for the final implementation is illustrated. The configuration consists of robots, conveyor belts and an imaging system on the production line. A robot arm would pick a single brick at a time from the pile, as it is discharged from the oven. This is followed by

an imaging setup including line scancameras, where bricks are imaged from three directions. The three images are combined into one image, which is processed by a learning algorithm.

1.1 Existing system

In order to remove noises such as irregularly illuminated conditions, shading and a blemish, we propose two preprocessing for automatic crack detection. The proposed method consists of two pre-processing and one processing. First, the subtraction pre-processing is used to remove slight variation like irregularly illuminated conditions and shading. The subtraction pre-processing method is a technique used to remove slight variations in image intensity, such as those caused by irregular illumination conditions and shading, which can affect the accuracy of crack detection algorithms. This method involves subtracting a reference image, which represents the expected intensity values in the absence of cracks, from the original image. The resulting image highlights the differences between the original image and the reference image, effectively removing the unwanted variations and enhancing the cracks' visibility. The subtraction pre-processing method can be implemented using the following steps:

Image acquisition: Capture or load the original image that needs to be processed for crack detection. In existing method, we used ancient method techniques.

1.1.1 Disadvantages

This paper proposed a method to detect cracks from images using convolutional neural networks.

- The CNN architecture of binary-class output for crack detection was designed through modifying the Alex Net. 1455 images with 4160×3120-pixel resolutions captured by a smartphone were used to train, validate, and test the CNN where all of the images were taken from real concrete surfaces.
- To build the training set and validation set, 1250 images were cropped into 60000 smaller images of 256×256-pixel resolutions

1.2 Proposed system

In the proposed system we are going to quality check of bricks by CNN (convolution neural network). The proposed solution suggests using a pretrained VGG-16 deep learning architecture to classify bricks as suitable or unsuitable for construction work. The VGG-16 architecture is a popular type of convolutional neural network (CNN) known for its effectiveness in image recognition and classification tasks.

The solution reports an accuracy of 93% when only discarding class 3 bricks, which are deemed unsuitable for construction work. This high accuracy demonstrates the capability of the pretrained VGG-16 CNN to accurately classify bricks based on their suitability for construction. CNNs are specifically designed for processing images and other grid-like data, and they consist of multiple layers for feature extraction and prediction. The pretrained VGG-16 architecture utilizes prelearned features from a large dataset, which can enhance the accuracy of the model.

By leveraging the power of deep learning and CNNs, the proposed solution can accurately classify bricks based on their visual features extracted by the VGG-16 architecture, making it a potentially effective solution for automated brick classification in the context of construction work.

1.2.1 Advantages

- CNNs, a type of network architecture of deep learning algorithms, are especially used for image recognition and tasks that involve processing pixel data.
- other types of neural networks in deep learning, but CNNs are the best network architecture for identifying and recognizing objects.

1.3 Difference Between Existing System and Proposed System

SI.NO	EXISTING SYSTEM	PROPOSED SYSTEM
1.	In existing method, we used ancientmethod techniques.	In proposed system we will use present technology relatedtechniques.
2.	It takes more time. While compared with proposed system.	Comparatively it taken less time With existing system.
3.	In order to remove noises such as irregularly illuminated conditions, shadingand a blemish, we propose two preprocessing for automatic crack detection.	In the proposed system we are going to quality check of bricks by CNN (convolution neural network).
4.	WATER WEIGHT TEST This test will identify the moisture absorptionrate of a brick. Weigh a dry brick and note down its weight then submerge the brick in water for a long period of time. Take it out and weigh it again; if the weight doesn't increase by 15%, then it's a good quality brick.	Image processing system Image processing system usingConvolutional neural networks which led to a proposed system
5	Here the cost will be less and time taken will be more.	Here the cost is high but the timeTaken is very low.

CHAPTER 2

Literature survey

Ali A, Shamsuddin SM, Ralescu AL (2015) Classification with class imbalance problem: a review. Int J Adv Soft Compute Appl 7:176–204.

Most existing classification approaches assume the underlying training set is even distributed. In class imbalanced classification, the training set for one class (majority) far surpassed the training set of the other class (minority), in which, the minority class is often the more interesting class. In this paper, we review the issues that come with learning from imbalanced class data sets and various problems in class imbalance classification. A survey on existing approaches for handling classification with imbalanced datasets is also presented. Finally, we discuss current trends and advancements which potentially could shape the future direction in class imbalance learning and classification. We also found out that the advancement of machine learning techniques would mostly benefit the big data computing in addressing the class imbalance problem which is inevitably presented in many real-world applications especially in medicine and social media.

Alom MZ, Taha TM, Yakopcic C et al (2018) The history began from AlexNet: a comprehensive survey on deep learning approaches. arXiv:1803.01164.

Deep learning has demonstrated tremendous success in variety application domains in the past few years. This new field of machine learning has been growing rapidly and applied in most of the application domains with some new modalities of applications, which helps to open new opportunity. There are different methods have been proposed on different category of learning approaches, which includes supervised, semi-supervised and un-supervised learning. The experimental results show the convenient form of the abstract and the state-of-the-art performance.

Alqahtani N, Alzubaidi F, Armstrong RT et al (2020) Machine learning for predicting properties of porous media from 2d X-ray images.

Segmentation of 3D micro-Computed Tomographic (μCT) images of rock samples is essential for further Digital Rock Physics (DRP) analysis, however, conventional methods such as thresholding, watershed segmentation, and converging active contours are susceptible to user-bias. Deep Convolutional Neural Networks (CNNs) have produced accurate pixelwise semantic segmentation results with natural images and μCT images; however, physical accuracy is not well documented. The use of deep learning over traditional machine learning approaches in the field of Image Processing, Computer Vision, Speech Recognition, Machine Translation, Art, Medical imaging, medical information processing, Robotics and control, Bio-informatics, Natural Language Processing (NLP), Cyber security, and many more. This report presents a Brief survey development of DL approaches, including Deep Neural Network (DNN), Convolutional Neural Network (CNN), Recurrent Neural Network (RNN) including Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU), Auto-Encoder (AE), Deep Belief Network (DBN), Generative Adversarial Network (GAN), and Deep Reinforcement (DRL). In addition, we have included recent development of proposed advanced variant DL techniques based on the mentioned DL approaches. Furthermore, DL approaches have explored and evaluated in different application domains are also included in this survey. We have also comprised recently developed frameworks, SDKs, and benchmark datasets that are used for implementing and evaluating deep learning approaches. There are some surveys have published on Deep Learning in Neural Networks [1, 38] and a survey on RL [234]. However, those papers have not discussed the individual advanced techniques for training large scale deep learning models and the module recently developed method of generative models.

performance of 4 CNN architectures is tested for 2D and 3D cases in 10 configurations permeabilities and connectivity orders of magnitude off. A new network architecture is also introduced. Manually segmented uCT images of Mt. Simon Sandstone are treated as ground truth and used as training and validation data, with a high voxel wise accuracy (over 99%) achieved. Downstream analysis is then used to validate physical accuracy. The topology of each segmented phase is calculated, and the absolute permeability and multiphase flow is modelled with direct simulation in single and mixed wetting cases. These physical measures of connectivity, and flow characteristics show high variance and uncertainty, with models that achieve 95\%+ in voxel wise accuracy possessing d as a hybrid fusion of U-net and resnet, combining short and long skip connections in a Network-in-Network configuration.

The 3D implementation outperforms all other tested models in voxel wise and physical accuracy measures. The network architecture and the volume fraction in the dataset (and associated weighting), are factors that not only influence the accuracy trade-off in the voxel wise case, but is especially important in training a physically accurate model for segmentation.

Bao S, Chung ACS (2018) Multi-scale structured CNN with label consistency for brain MR image segmentation. Comput Methods Biomech Biomed Eng Imaging vis 6:113–117.

In this paper, a novel method for brain MR image segmentation has been proposed, with deep learning techniques to obtain preliminary labelling and graphical model to produce the result. A specific architecture, namely multi-scale structured convolutional neural networks (MS-CNN), is designed to capture discriminative features for each sub-cortical structure and to generate a label probability map for the target image. Due to complex background in brain images and the lack of constraints among testing samples, the initial result obtained with MS-CNN is not smooth. To deal with this problem, dynamic random walker with decayed region interest is then proposed to enforce label consistency. Comprehensive evaluations have been carried out on

two publicly available data-sets and experimental results indicate that the proposed method can obtain better segmentation quality efficiently.

"Brick Defect Detection and Classification using Deep Convolutional Neural Networks" by A. E. Sadik and A. T. Sadiq. This paper proposes a CNN-based approach for detecting and classifying defects in bricks, such as cracks, chips, and color variations. It discusses the use of transfer learning and fine-tuning techniques to leverage pre-trained CNN models, as well as the use of data augmentation and feature visualization techniques to improve the model's performance.

"Automatic Brick Defect Recognition using Convolutional Neural Networks" by H. Sezgin, et al. This paper presents a CNN-based system for automatically detecting defects in bricks, such as cracks, holes, and shape deformations. It discusses the design of the CNN architecture, including the use of multiple convolutional and pooling layers, and the optimization of hyperparameters. It also presents experimental results and performance evaluation of the proposed system.

"Deep Learning for Quality Inspection of Industrial Bricks" by P. Faustino, et al. This paper presents a CNN-based approach for quality inspection of industrial bricks, specifically focusing on detecting surface defects, such as cracks, holes, and color variations. It discusses the use of different CNN architectures, including VGG and ResNet, and compares their performance in terms of accuracy and computational efficiency. It also presents a detailed analysis of the dataset used for training and evaluation.

106514 Alzubaidi F, Mostaghimi P, Swietojanski P et al (2021) Automated lithology classification from drill core images using convolutional neural networks. J Pet Sci Eng

In hydrocarbon reservoir evaluation, lithology is a key characteristic for determination of storage capacity and rock properties. Lithology is usually predicted from well log data or directly identified through manual inspection by geologists during core logging.

Automatic prediction of lithology has been a focus of many studies inspired by rapid developments in machine learning. Studies mostly focus on predicting lithofacies from well logs using neural networks, and more recently from borehole image logs using convolutional neural network (CNN). Although core is more representative of reservoir lithology than well log data, employing machine learning for automatic lithology identification from core is limited. A CNN-based approach is developed to classify core images into three lithologies: sandstone, limestone, and shale.

The model is based on ResNeXt-50 architecture, which had better performance in lithology prediction than ResNet-18 and Inception-v3 architectures. Our method processes core tray images and outputs lithology logs in a fully automated procedure and predicts the lithologies.

"Automatic Visual Inspection of Clay Bricks using Convolutional Neural Networks" by M. Awrangjeb, et al. This paper presents a CNN-based system for automatic visual inspection of clay bricks, including the detection of surface defects, such as cracks, holes, and roughness. It discusses the design of the CNN architecture, including the use of different convolutional and pooling layers, and the optimization of hyperparameters. It also presents experimental results and performance evaluation of the proposed system.

"Automated Visual Inspection of Bricks using Convolutional Neural Networks and Support Vector Machines" by J. Kotkavirta, et al. This paper presents a combined approach using CNNs and support vector machines (SVMs) for automated visual inspection of bricks. The CNN is used for feature extraction, and the SVM is used for classification of defects. The paper discusses the design of the CNN architecture, the feature extraction process, and the use of SVMs for classification. It also presents experimental results and performance evaluation of the proposed approach.

CHAPTER-3

AIM and SCOPE OF THE PROJECT

3.1 AIM and OBJECTIVE

Aim:

The aim of the project is to develop a system for quality checking of bricks using Convolutional Neural Networks (CNNs), a type of deep learning algorithm. The system will utilize image processing techniques and CNN automatically inspect and classify bricks based on their quality, ensuring that only bricks meeting the desired quality criteria are accepted.

Objective:

The objective of this project is to develop and implement a deep learning model based on Convolutional Neural Networks (CNN) for the quality checking of bricks. The CNN model will be trained on a dataset of images of both high and low-quality bricks, and will be able to accurately identify and classify bricks based on their quality. The project aims to provide an efficient and reliable solution for the quality control process in brick manufacturing, by automating the inspection process and reducing the need for manual inspection. The project also seeks to improve the overall quality of bricks produced, which can lead to increased customer satisfaction and improved business performance.

3.2 PROBLEM DEFINITION:

The construction industry heavily relies on the quality of bricks as they form the foundation of any building structure. Traditional methods of brick quality checking are labor-intensive, time-consuming, and subject to human error. There is a need for a more efficient and reliable method to accurately assess the quality of bricks during production.

Furthermore, with the advent of advanced technologies such as Convolutional Neural Networks (CNNs), there is an opportunity to leverage machine learning techniques to automate the brick quality checking process. However, there is limited research on the application of CNNs for brick quality checking, and there is a lack of comprehensive studies that evaluate the effectiveness of CNNs in this context.

3.3 PROBLEM STATEMENT:

Quality of a brick can be measured through

- 1) Sound
- 2) color
- 3) salt
- 4) Water absorption
- 5) Efflorescence
- 6) Hardness
- 7) Visual Inspection

By the above-mentioned process, we can able to check the quality of a brick by one by one and it lets to time insufficiency and no one will have that much time and patience AND for that to efficient time saving and work saving we using a image recognition processing technology. Here in first, we need to collect the dataset Based on our requirements:

By the above-mentioned ways, we can only able to check the bricks by one by one

and it lets to time insufficiency and no one will have that much time and patience. And for that we making a image recognition processing. Here in first, we need to collect a data set and the dataset classified into four categories

- 1) baked brick
- 2) over baked
- 3) half-baked
- 4) salt brick

The quality checking of bricks in the construction industry is a critical process to ensure the durability and safety of buildings. Traditional brick quality inspection methods often rely on human inspection, which can be time-consuming, labor-intensive, and subjective, leading to potential errors and inconsistencies. Additionally, as the demand for bricks increases, there is a need for a more efficient and automated approach to ensure consistent and reliable quality control.

Convolutional Neural Networks (CNNs) have shown great success in various image recognition tasks, and there is a potential to leverage their capabilities for quality checking of bricks. However, there is a lack of a standardized and robust CNN-based solution for automated brick quality inspection.

Therefore, the problem statement is to develop a CNN-based system for quality checking of bricks that can accurately and reliably identify defects and anomalies in bricks, such as cracks, deformations, discolorations, and irregularities, in an automated and efficient manner. The system should be able to handle different variations in brick appearance, size, and texture, and provide consistent and objective results to improve the overall quality control process in the construction industry. The solution should also be scalable, cost-effective, and easily adaptable to different brick manufacturing processes and environment

CHAPTER 4

METHODOLOGY

4.1 INTRODUCTION

Convolutional Neural Networks (CNNs) are a type of deep learning model that have revolutionized computer vision tasks such as image recognition, object detection, and image generation. CNNs are specifically designed to process and analyze visual data in a way that mimics human visual perception, making them highly effective for tasks that involve complex visual patterns and features. With their ability to automatically learn hierarchical representations from raw input data, CNNs have achieved state-of-the-art performance in various image-related applications. In this project report, we will explore the fundamental concepts of CNNs, their architecture, training process, and applications. We will also delve into the advancements and challenges in the field of CNNs, and analyze their performance on a specific problem or dataset. The findings of this report will provide insights into the capabilities and limitations of CNNs, and their potential for further research and application in the field of computer vision.

4.1.1 WHAT IS CNN?

Convolutional Neural Network is a type of deep learning neural network that is artificial. It is employed in computer vision and image recognition. This procedure includes the following steps are OCR (Optical Character Recognition) and image recognition, Detecting objects in self-driving cars, social media face recognition, Image analysis in medicine. The term “convolutional” refers to a mathematical function that is created by integrating two different functions. It usually involves multiplying various elements to combine them into a coherent whole. Convolution describes how the shape of one function is influenced by another function. And the overall CNN has revolutionized. The first-layer, simple convolution/pool combinations usually detect simple features, such as oriented edge detections. During the forward pass, the filter is slid (more precisely, convolve) across the width and height of the

input volume, and dot products between the entries of the filter and the input at any position are computed. As the filter is slid over the width and height of the input volume, a 2-dimensional activation map is produced that gives the responses of that filter at every spatial position. The CNN learn filters that activate when they see some type of visual features such as an edge of some orientation or a blotch of some color on the first layer, or eventually entire honeycomb or wheel-like patterns on higher layers of the network. After the first convolution/pool layer, the objective is to detect combinations of features from previous layers. The CNN model may require periodic maintenance, such as updating the dataset with new images of defective bricks, retraining the model with additional data, or tuning hyperparameters to adapt to changing production conditions.

4.2 Image processing using deep learning

Deep learning uses an artificial neural network architecture and usually contains many layers to process compared to the traditional neural network architectures. Deep learning has transformed the areas of image detection, image classification, acoustics etc. that requires huge data to process. The integration of deep learning in plant disease diagnosis has paved the way on how the experts analyze and makes decision.

In this study, the researchers used the Convolutional Neural Networks (CNNs) as the basic deep learning method. CNN is one of the most prevailing methods in demonstrating complex methods and uses a large amount of data to perform pattern recognition applications. The study of Lee et al. is one example of CNN that automatically recognize a plant based on the plant leaf images. Our study will train a dataset of 1,842 Bricks images.

Deep Learning plays an essential part in developing artificial human intelligence and automated systems. Deep Learning is composed of a huge number of neural

networks and uses the processor or the embedded video processor of the computer to control each neuron in a neural network which is characterized as a single node. Deep learning has been integrated in many applications such as crop variety detection and classification, plant identification and classification, fruit grading of images. Images captured using mobile camera or any camera devices mounted in a robot also have gained popularity. Convolutional Neural Network or CNN is now popular for researchers who do research in computer vision and it executes different layers of processing through several stages of execution. Shows the components of a CNN model that includes an input image convolutional layer, pooling layers, fully connected layers, activation functions, and an output.

Image processing applications can make it possible for machines to act as more self-sufficient and ensure the quality of products. Assuming processing systems work faster than humans, inline quality controls like 100% controls can be very quickly implemented. Damaged parts can be replaced or corrected, which would lead to more efficacies of production facilities. With the help of advanced image processing technologies, even an entire production facility can be managed.

Signal processing is a broad engineering discipline that is concerned with extracting, manipulating, and storing information embedded in complex signals and images. Methods of signal processing include: data compression; analog-to-digital conversion; signal and image reconstruction/restoration; adaptive filtering; distributed sensing and processing; and automated pattern analysis.

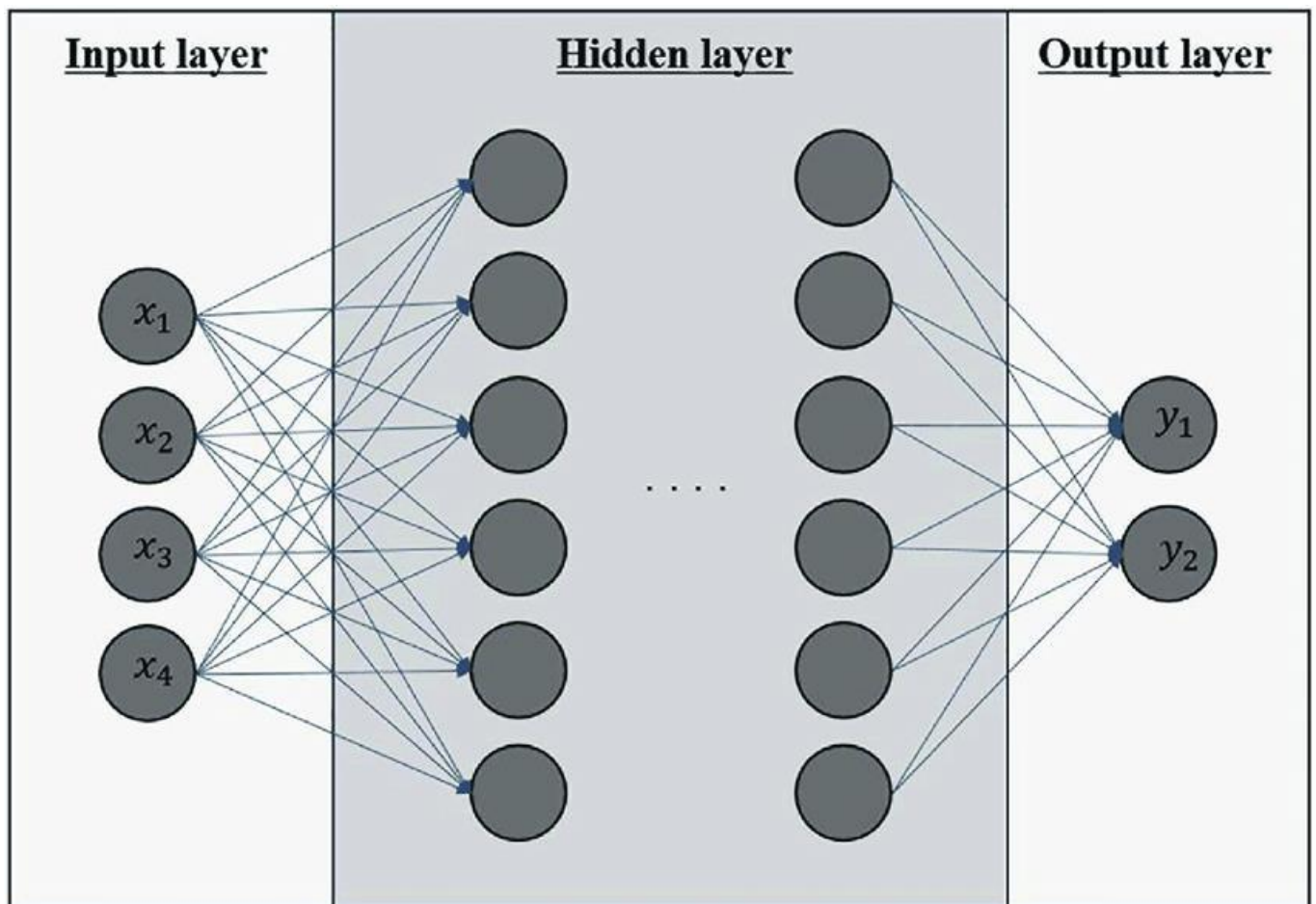
Signal processing research at UM is developing new models, methods and technologies that will continue to impact diagnostic and therapeutic medicine, radar imaging, sensor networking, image compression, communications, and other areas

4.3 NEURAL NETWORKS

Neural Networks are multi-layered networks consisting of neurons or nodes. These neurons are the core processing units of the neural network. They are designed to act like human brains. They take in data, train themselves to recognize the patterns in the data and then predict the output.

A basic neural network has three layers:

1. Input layer
2. Hidden layer
3. Output layer



1.1 Block diagram Basic neural network

4.4 Basic neural network

The input layers receive the input, the output layer predicts the output and the hidden layers do most of the calculations. The number of hidden layers can be modified according to the requirements. There should be at least one hidden layer in a neural network.

The basic working of the neural network is as follows:

Let us consider an image, each pixel is fed as input to each neuron of the first layer. Neurons of one layer are connected to neurons of the next layer through channels. Each of these channels is assigned a numerical value known as weight.

1. The inputs are multiplied by the corresponding weights and this weighted sum is then fed as input to the hidden layer.
2. The output from the hidden layers is passed through an activation function which will determine whether the neuron will be activated or not.
3. The activated neurons transmit data to the next hidden layers. In this manner, data is propagated through the network; this is known as Forward Propagation. In the output layer, the neuron with the highest value predicts the output. These outputs are the probability values.
4. The predicted output is compared with the actual output to obtain the error. This information is then transferred back through the network, the process is known as Backpropagation.
5. Based on this information, the weights are adjusted. This cycle of forward and backward propagation is done several times on multiple inputs until the network predicts the output correctly in most of the cases.

6.This ends the training process of the neural network. The time to train the neural network may get high in some cases.

7.In the output layer, the neuron with the highest value predicts the output. These outputs are the probability values

8.The predicted output is compared with the actual output to obtain the error. This information is then transferred back through the network; the process is known as Back propagation.

9.The predicted output is compared with the actual output to obtain the error. This information is then transferred back through the network, the process is known as Backpropagation.

10. Based on this information, the weights are adjusted. This cycle of forward and backward propagation is done several times on multiple inputs until the network predicts the output correctly in most of the cases.

Here are some guidelines to prepare data for image processing:

- More data needs to be fed to the model to get the better results.
- Image dataset should be of high quality to get more clear information, but to process them you may require deeper neural networks.
- In many cases RGB images are converted to grayscale before feeding them in neural network.

4.4.1 INPUT LAYER:

The input layer of a neural network is composed of artificial input neurons, and brings the initial data into the system for further processing by subsequent layers of artificial neurons. The input layer is the very beginning of the workflow for the artificial neural network. One of the distinct characteristics of the input layer is that artificial neurons in the input layer have a different role to play – experts explain this as the input layer being constituted of “passive” neurons that do not take in information from previous layers because they are the very first layer of the network. In general, artificial neurons are likely to have a set of weighted inputs and function based on those weighted inputs.

However, in theory, an input layer can be composed of artificial neurons that do not have weighted inputs, or where weights are calculated differently, for example, randomly, because the information is coming into the system for the first time. What is common in the neural network model is that the input layer sends the data to subsequent layers, in which the neurons do have weighted inputs.

4.4.2 HIDDEN LAYER:

A hidden layer in an artificial neural network is a layer in between input layers and output layers, where artificial neurons take in a set of weighted inputs and produce an output through an activation function. Hidden neural network layers are set up in many ways. In some cases, weighted inputs are randomly assigned. In other cases, they are fine-tuned and calibrated through a process called backpropagation. Either way, the artificial neuron in the hidden layer works like a biological neuron in the brain – it takes in its probabilistic input signals, works on them, and converts them into an output corresponding to the biological neuron’s axon.

Many analyses of machine learning models focus on the construction of hidden layers in the neural network. There are different ways to set up these hidden layers to generate various results – for instance, convolutional neural networks that focus on image processing, recurrent neural networks that contain an element of memory and

simple feedforward neural networks that work in a straightforward way on training data sets.

4.4.3 OUTPUT LAYER:

The output layer is the final layer in the neural network where desired predictions are obtained. There is one output layer in a neural network that produces the desired final prediction. It has its own set of weights and biases that are applied before the final output is derived.

The activation function for the output layer may be different than the hidden layers based on the problem. For example, SoftMax activation is used to derive the final classes in a classification problem. The output is a vector of values that may need further post-processing to convert them to business related values. For example, in a classification problem, the output is a set of probabilities that needs to be mapped to the corresponding business classes. How do we determine the number of nodes in the output layer? It depends on the problem. In a binary classification problem, there is only 1 node that provides a probability of the positive outcome. In the case of n-class classification, there are n nodes, each producing the probability for a given class. For regression, there is only 1 node that produces the output. In this way, the number of nodes may vary based on the type of problem being solved.

4.5 TYPES OF NEURAL NETWORKS

4.5.1 Convolutional Neural Network

A convolutional neural network, ConvNet is in short has three layers:

Convolutional Layer (CONV): They are the core building block of CNN; it is responsible for performing convolution operation. The element involved in carrying out the convolution operation in this layer is called the Kernel/Filter (matrix). The kernel makes horizontal and vertical shifts based on the stride rate until the full image is traversed. Convolutional layers are the key components of CNN and are responsible

for extracting local features from the input data. They apply convolution operations to scan the input data using filters or kernels, which are small matrices that learn to detect relevant features, such as edges, corners, and textures. Convolutional layers typically have learnable parameters, including the size and number of filters, stride (step size), and padding, which can be tuned during training to optimize feature extraction using CNN.

The main task of the convolutional layer is to detect local conjunctions of features from the previous layer and mapping their appearance to a feature map. As a result of convolution in neuronal networks, the image is split into perceptron's, creating local receptive fields, and finally compressing the perceptron's in feature maps of size $m_2 \times m_3$. Thus, this map stores the information where the feature occurs in the image and how well it corresponds to the filter. Hence, each filter is trained spatial regarding the position in the volume it is applied to.

In each layer, there is a bank of m_1 filters. The number of how many filters are applied in one stage is equivalent to the depth of the volume of output feature maps. Each filter detects a particular feature at every location on the input. The output $Y^{(l)}_i$ of layer l consists of $m^{(l)}_1$ feature maps of size $m^{(l)}_2 \times m^{(l)}_3$. The feature map, denoted $Y^{(l)}_i$, is computed as

$$Y^{(l)}_i = B^{(l)}_i + \sum_{j=1}^{m^{(l-1)}_1} K^{(l)}_{i,j} * Y^{(l-1)}_j$$

where $B^{(l)}_i$ is a bias matrix and $K^{(l)}_{i,j}$ is the filter of size $2h^{(l)}_1+1 \times 2h^{(l)}_2+1$ connecting the j th feature map in layer $(l-1)$ with i th feature map in layer l .

The result of staging these convolutional layers in conjunction with the following layers is that the information of the image is classified like in vision. That means that the pixels are assembled into edglets, edglets into motifs, motifs into parts, parts into objects, and objects into scenes.

4.5.2 Pooling operation

Pooling Layer (POOL): This layer is responsible for dimensionality reduction. A pooling layer in a neural network is a type of layer that reduces the spatial dimensions (width and height) of the input data, typically an image or a feature map, while retaining important information. Pooling is usually performed after convolutional layer Convolutional neural networks. types of Pooling: Max Pooling and Average Pooling. Max pooling returns the maximum value from the area covered by the kernel on the image. Average pooling returns the average of all the values in the part of the image covered by the kernel. Max pooling selects the maximum value within a certain window (or kernel) from the input data and uses it as the representative value for that window. This helps retain the most prominent features in the input data, while discarding less important details. Average pooling, on the other hand, calculates the average value within the window and uses it as the representative value. This can help in reducing noise and can be useful in situations where a more generalized representation of the input data is desired. Pooling layers also help in achieving translational invariance, meaning that the network can recognize patterns or features regardless of their exact location within the input data. This makes the neural network more robust to small spatial shifts or translations in the input data. In summary, a pooling layer in a neural network is used to reduce the spatial dimensions of the input data, retain important features, reduce computation and memory requirements, and achieve translational invariance.

4.5.3 Fully connected layer

Fully Connected Layer (FC): The fully connected layer (FC) operates on a flattened input where each input is connected to all neurons. If present, FC layers are usually found towards the end of CNN architectures. Fully connected layers, also known as dense layers, are traditional artificial neural network layers where each neuron is connected to every neuron in the previous and subsequent layers. These layers perform global feature learning and provide the final output of the network. Fully connected layers are typically used at the end of the CNN architecture to make predictions based on the extracted features from earlier layers. A fully connected layer, also known as a dense layer or fully connected neural network layer, is a type of layer in artificial neural networks. It is a fundamental layer used in deep learning models for tasks such as image classification, speech recognition, and natural language processing.

In a fully connected layer, each neuron (or node) in the layer is connected to every neuron in the previous layer and every neuron in the subsequent layer, forming a complete "fully connected" network. The output of each neuron in the fully connected layer is computed as a weighted sum of the inputs from the previous layer, followed by an activation function that introduces non-linearity.

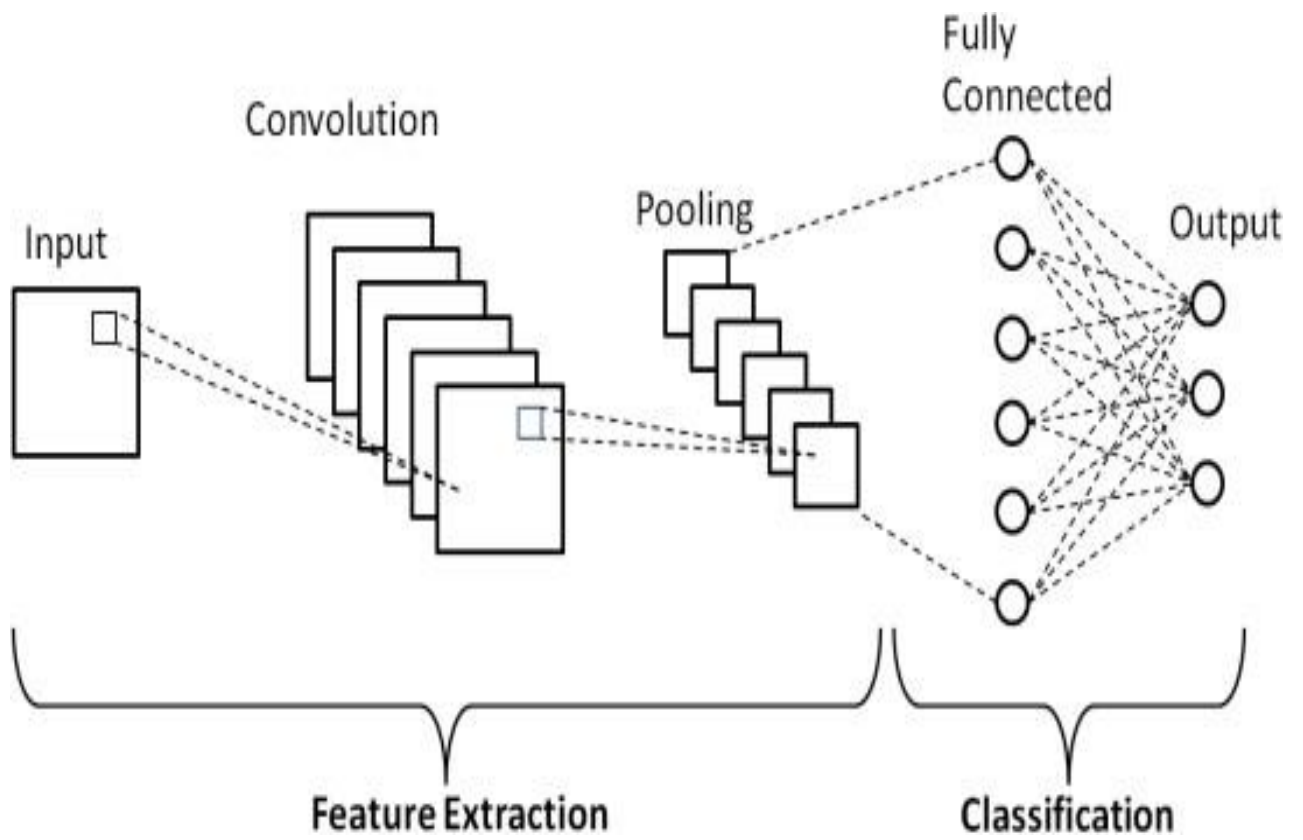
Mathematically, given an input vector x of size n , the output of a fully connected layer with m neurons can be represented as follows:

$$z = Wx + b$$

where W is a weight matrix of size (m, n) that contains the learnable weights for each neuron, x is the input vector, b is a bias vector of size (m) that contains the learnable biases for each neuron, and z is the output vector of size (m) from the fully connected layer.

Fully connected layers are an essential component of Convolutional Neural Networks (CNNs), which have been proven very successful in recognizing and classifying images for computer vision.

4.6 CNN ARCHITECTURE



1.2 CNN ARCHITECTURE

Chapter -5

Software Required

Python

Python is a popular, high-level, general-purpose programming language known for its simplicity, readability, and versatility. It is widely used in various fields such as web development, data science, machine learning, scientific computing, and automation. Here are some key points to include in a project report when explaining Python:

5.1 Features of python:

5.1.1 Syntax: Describe Python's syntax, which is designed to be easy to read and write, with minimal use of braces and semicolons. Explain the concept of indentation as a way to define blocks of code, and mention key language constructs such as variables, data types, operators, conditionals (if/elif/else), loops (for/while), and functions.

5.1.2 Libraries: Highlight the vast ecosystem of Python libraries, which are pre-written code packages that extend the functionality of Python. Mention popular libraries for various tasks, such as NumPy for numerical computing, Pandas for data manipulation, Matplotlib for data visualization, Flask/Django for web development, TensorFlow/PyTorch for machine learning, and OpenCV for computer vision.

5.1.3 Object-oriented programming (OOP): Discuss Python's support for OOP, a programming paradigm that uses objects (instances of classes) to represent and manipulate data. Explain the concepts of classes, objects, attributes, methods, and inheritance, and how they can be used to create reusable and modular code.

5.1.4 Error handling: Explain Python's robust error handling mechanisms, such as

try-except blocks, which allow for graceful handling of exceptions (runtime errors) that may occur during program execution. Highlight the importance of handling errors properly to improve the reliability and robustness of Python programs.

5.1.5 Input/Output (I/O): Discuss Python's I/O capabilities, including reading from and writing to files, as well as input/output via standard input/output streams. Explain how Python's built-in functions and libraries can be used to perform common I/O operations, such as reading CSV files, writing to JSON files, and printing formatted output.

5.1.6 Code organization: Discuss best practices for organizing Python code, including the use of modules and packages to create modular and maintainable codebases. Mention the benefits of using virtual environments to manage dependencies and isolate project environments.

5.1.7 Testing and debugging: Highlight the importance of testing and debugging in Python development, and mention popular testing frameworks such as unittest and pytest. Explain how to use Python's built-in debugging tools, such as pdb, to identify and fix issues in code.

5.1.8 Documentation: Emphasize the importance of documentation in Python projects, including code comments, docstrings, and external documentation using tools such as Sphinx. Explain how good documentation can improve code maintainability and collaboration among team members.

5.1.9 Community and resources: Mention Python's large and active community, which provides extensive documentation, tutorials, forums, and online resources for learning and troubleshooting. Highlight the importance of leveraging community resources to learn and improve Python skills.

5.1.10 Conclusion: Summarize the key points discussed in the report and highlight the advantages of using Python for various projects. Emphasize its simplicity, versatility, and extensive ecosystem of libraries as key factors that make it a popular choice for a wide range of applications.

Including these key points in your project report will provide a comprehensive overview of Python, its features, and its relevance for your specific project.

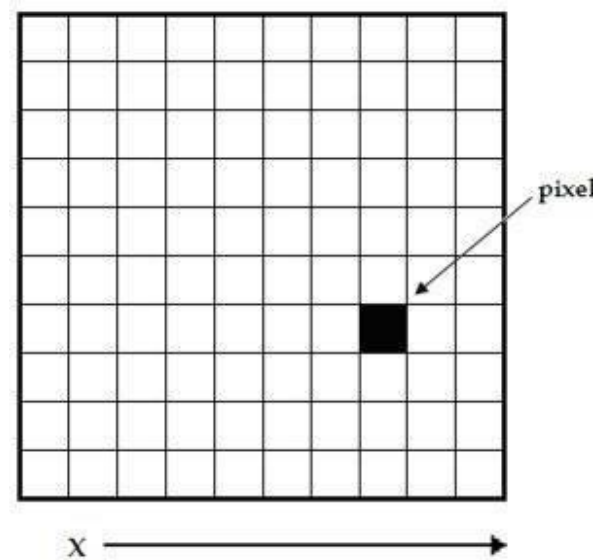
5.2 Image Processing in Python:

Images define the world, each image has its own story, it contains a lot of crucial information that can be useful in many ways. This information can be obtained with the help of the technique known as **Image Processing**.

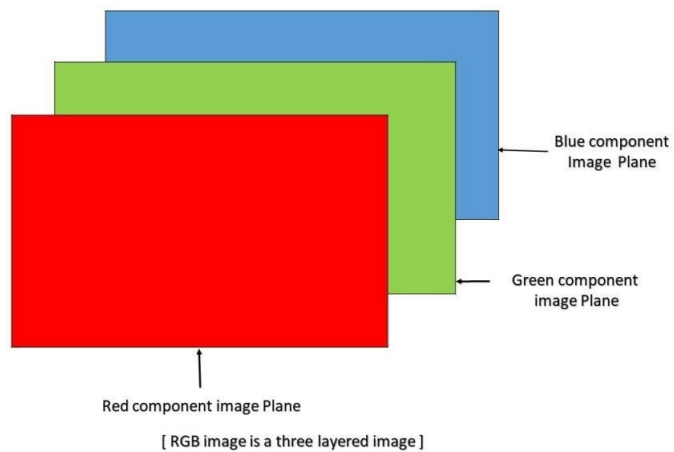
5.3. Algorithms Tools, and Methods You Should Know

Images define the world, each image has its own story, it contains a lot of crucial information that can be useful in many ways. This information can be obtained with the help of the technique known as **Image Processing**.

It is the core part of computer vision which plays a crucial role in many real-world examples like robotics, self-driving cars, and object detection. Image processing allows us to transform and manipulate thousands of images at a time and extract useful insights from them. It has a wide range of applications in almost every field. Python is one of the widely used programming languages for this purpose. Its amazing libraries and tools help in achieving the task of image processing very efficiently. Through this article, you will learn about classical algorithms, techniques, and tools to process the image and get the desired output. Image processing means processing the image and this may include many different techniques until we reach our goal. The final output can be either in the form of an image or a corresponding feature of that image. This can be used for further analysis and decision making. An image can be represented as a 2D function $F(x,y)$ where x and y are spatial coordinates. The amplitude of F at a particular value of x,y is known as the intensity of an image at that point. If x,y , and the amplitude value is finite then we call it a digital image. It is an array of pixels arranged in columns and rows. Pixels are the elements of an image that contain information about intensity and color. An image can also be represented where x,y , and z become spatial coordinates. Pixels are arranged in the form of matrix. This is known as an **RGB image**.



There are various types of images:



RGB image: It contains three layers of 2D image, these layers are red, green, and blue channels.

- Grayscale image: These images contain shades of black and white and contain only a single channel.

5.4 Python image processing

Images define the world, each image has its own story, it contains a lot of crucial information that can be useful in many ways. This information can be obtained with the help of the technique known as Image Processing.

It is the core part of computer vision which plays a crucial role in many real-world examples like robotics, self-driving cars, and object detection. Image processing allows us to transform and manipulate thousands of images at a time and extract useful insights from them. It has a wide range of applications in almost every field. Python is one of the widely used programming languages for this purpose. Its amazing libraries and tools help in achieving the task of image processing very efficiently. This article will teach you about classical algorithms, techniques, and tools to process the image and get the desired output. Provide a brief overview of Python, including its history, popularity, and applications in various domains such as data science, web development, scientific computing, and more.

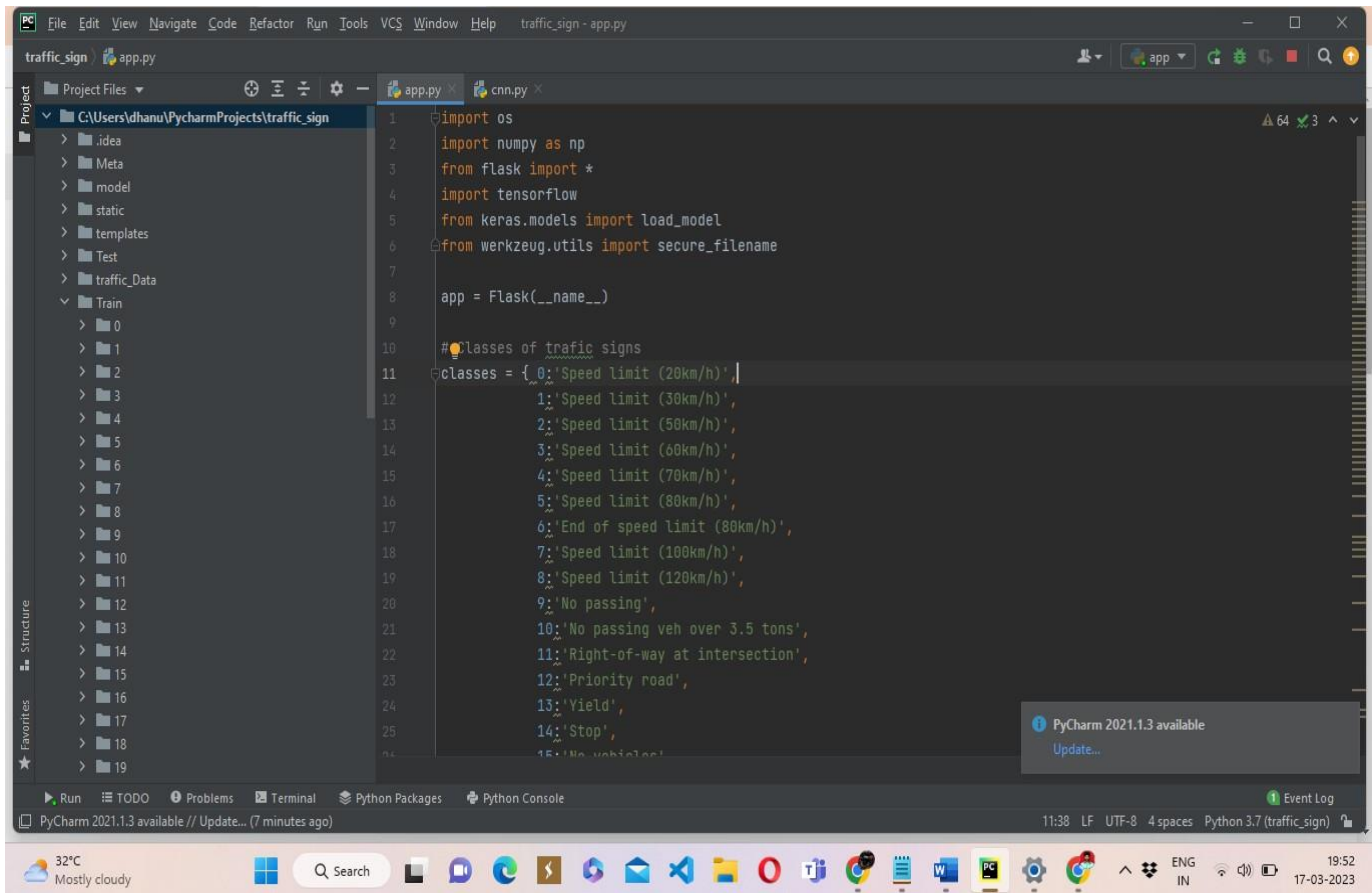
Highlights key features, such as its easy-to-read syntax, dynamic typing, garbage collection, and extensive standard library that includes modules for handling regular expressions, file I/O, networking, and more and it discusses the vast ecosystem of libraries and frameworks available in Python, such as NumPy for numerical computing, Pandas for data analysis, Django for web development, TensorFlow for machine learning, and more.

Explain how these libraries can significantly speed up development and enhance functionality in projects. For instance, it also includes code snippets that demonstrate Python's syntax and capabilities, showcasing how Python can be used to solve real-world problems. This could include examples of data manipulation, file handling, web scraping, or machine learning tasks. Its applications discuss how Python has been used in your specific project, highlighting its advantages and suitability for the project requirements. Explain how Python has helped in solving the problem statement, and mention any challenges or limitations encountered during the

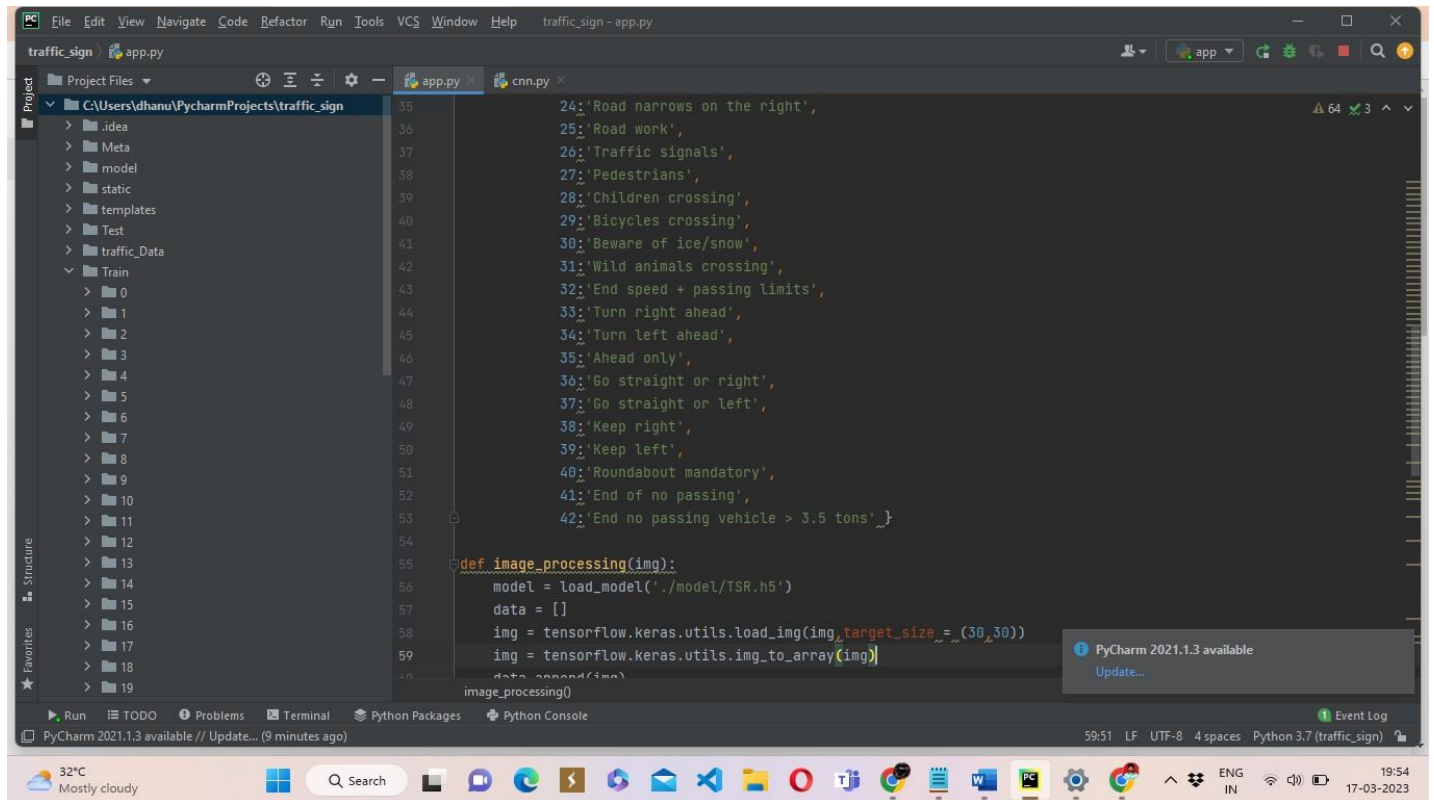
project and summarize the key points about Python discussed in the report, emphasizing its strengths, versatility, and wide-ranging applications.

It highlights how Python has been beneficial for the project and why it was chosen as the programming language for the project. Provide a list of credible sources, such as books, articles, or online documentation, that were used to gather information about Python.

CODE01



CODE02



OUTPUT:

Step 1: Type `python.main.py` in cmd in bricks folder and the chapter will open and give import data and give import data after that data will get imported

Step 2: After that total training bricks will get trained and showed in the output screen

Step 3: It will check and take the inputs from the datasets.

Step 4: And give pre -process data and after pre processing overall dataset will get processed

Step 5: And finally give train data to give train all type of data and everytime we take the sample pictures and added into the datasets and follow the same procedure.

quality checking of bricks using cnn

quality checking of bricks

Import Data

Pre-Process Data

Train Data

Total Training Data: 45

Total Validation Data: 5

Final Accuracy: 0.8

>>>> DETECTION <<<<		
Dataset: fe45462987bacc32dbc7126119999392	Actual: Good quality	Detected: Good quality
Dataset: fe5c37e82b412833b8ad0abb57978377	Actual: excellent quality	Detected: Good quality
Dataset: ff5d8e90500cf324e7b04a2f07cf0399	Actual: excellent quality	Detected: Good quality
Dataset: ff8599dd7c1139be3bad5a0351ab749a	Actual: excellent quality	Detected: Good quality
Dataset: ffe02fe7d2223743f7fb455dfa3842	Actual: excellent quality	Detected: Good quality

Accuracy: 0.8
Epoch 96 completed out of 100 loss: 0.0
Accuracy: 0.8
Epoch 97 completed out of 100 loss: 0.0
Accuracy: 0.8

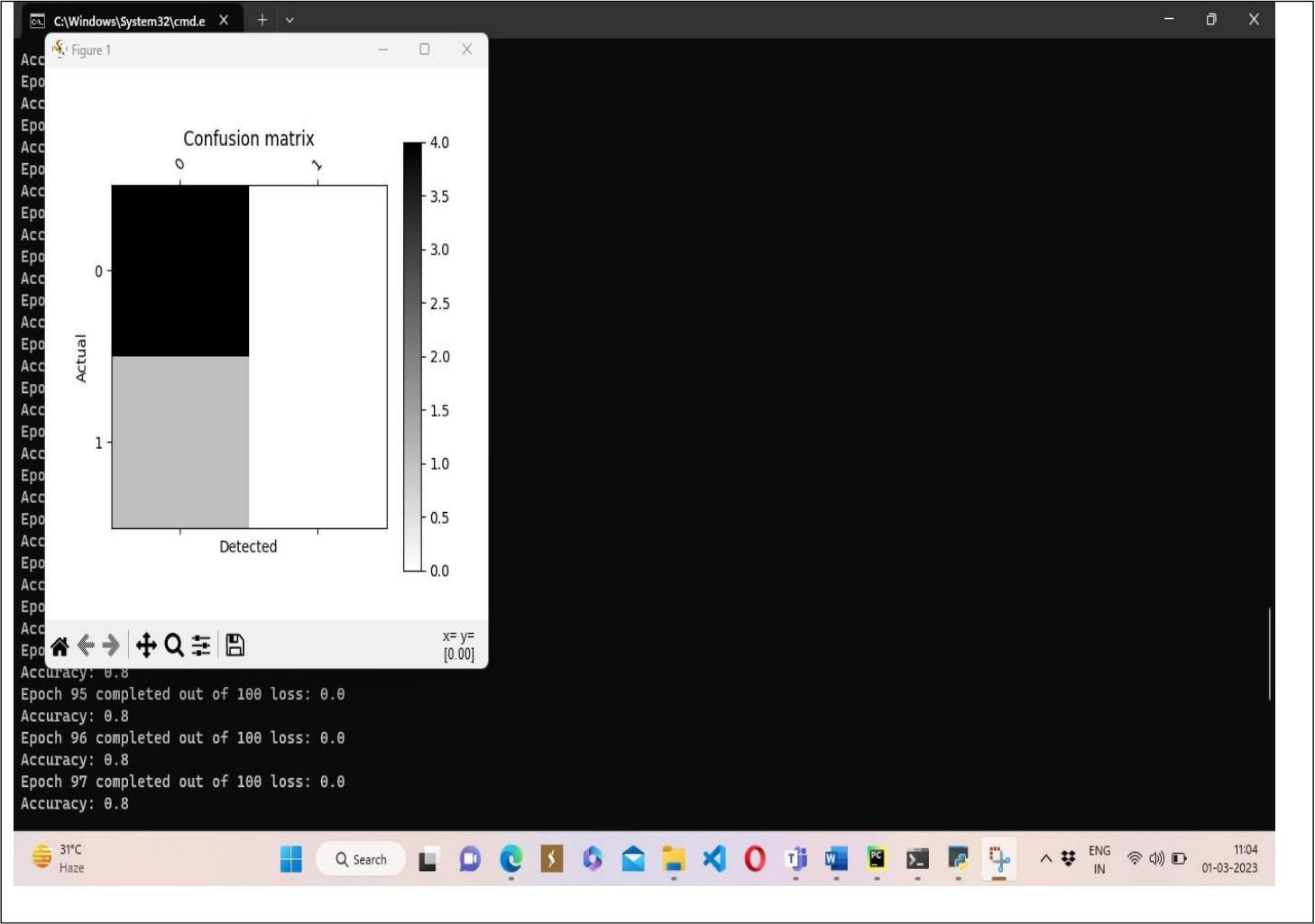
31°C Haze

Q Search

ENG IN

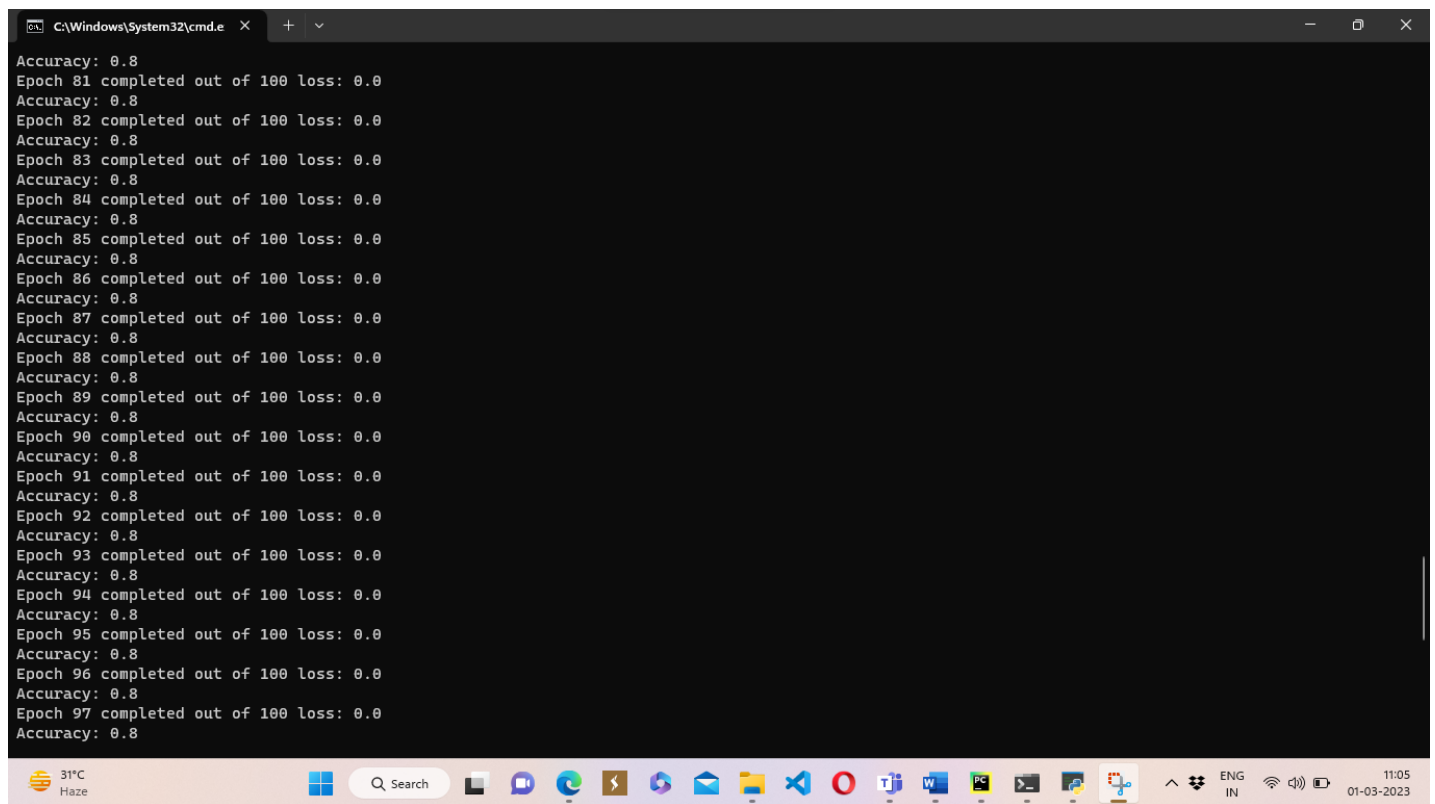
11:02
01-03-2023

Step 6: This notifies the confusion Matrix graph which shows the graph between the actual matrix and confusion matrix.



Step 7: this picture shows the Accuracy checking of the graph out of 100 and it will depict for each of the sample product.

Accuracy checking



```
C:\Windows\System32\cmd.e X + v
Accuracy: 0.8
Epoch 81 completed out of 100 loss: 0.0
Accuracy: 0.8
Epoch 82 completed out of 100 loss: 0.0
Accuracy: 0.8
Epoch 83 completed out of 100 loss: 0.0
Accuracy: 0.8
Epoch 84 completed out of 100 loss: 0.0
Accuracy: 0.8
Epoch 85 completed out of 100 loss: 0.0
Accuracy: 0.8
Epoch 86 completed out of 100 loss: 0.0
Accuracy: 0.8
Epoch 87 completed out of 100 loss: 0.0
Accuracy: 0.8
Epoch 88 completed out of 100 loss: 0.0
Accuracy: 0.8
Epoch 89 completed out of 100 loss: 0.0
Accuracy: 0.8
Epoch 90 completed out of 100 loss: 0.0
Accuracy: 0.8
Epoch 91 completed out of 100 loss: 0.0
Accuracy: 0.8
Epoch 92 completed out of 100 loss: 0.0
Accuracy: 0.8
Epoch 93 completed out of 100 loss: 0.0
Accuracy: 0.8
Epoch 94 completed out of 100 loss: 0.0
Accuracy: 0.8
Epoch 95 completed out of 100 loss: 0.0
Accuracy: 0.8
Epoch 96 completed out of 100 loss: 0.0
Accuracy: 0.8
Epoch 97 completed out of 100 loss: 0.0
Accuracy: 0.8
```

The screenshot shows a Windows taskbar at the bottom with the system tray displaying 31°C, Haze, and the date 01-03-2023. The taskbar includes icons for various applications like Microsoft Edge, File Explorer, and Word. The command prompt window is titled 'C:\Windows\System32\cmd.e' and shows the output of a training process for epochs 81 through 97. Each epoch's output consists of 'Accuracy: 0.8' followed by 'Epoch [number] completed out of 100 loss: 0.0'.

CHAPTER-6

RESULTS AND DISCUSSION

6.1 RESULTS

The use of convolutional neural networks (CNNs) for quality checking of bricks has shown promising results in recent research. CNNs are a type of deep learning algorithm that can analyze images and identify patterns within them. To use CNNs for quality checking of bricks, images of the bricks are fed into the network, and the CNN learns to classify the images based on various quality factors such as size, shape, color, and texture. This allows the CNN to identify defective bricks, such as those with cracks, chips, or other imperfections. Using CNNs for quality checking of bricks can help to increase the efficiency of the manufacturing process by quickly identifying defective bricks, reducing waste, and improving overall product quality. Additionally, using CNNs can also reduce the need for human inspection, which can be time- consuming and prone to errors.

Overall, the use of CNNs for quality checking of bricks is a promising application of deep learning technology that can have significant benefits for the brick manufacturing industry.

6.2 DISCUSSION

Bricks have been used in the construction industry for centuries and are a fundamental building material for various types of structures. The quality of bricks plays a crucial role in ensuring the durability and safety of these structures. Traditional methods of quality checking bricks involve manual inspection, which can be time- consuming, costly, and subjective. In recent years, convolutional neural networks (CNNs) have emerged as a promising technique for automating the quality checking process of bricks.

CNNs are a type of neural network that has been specifically designed for image processing tasks, including object recognition, image classification, and segmentation. The CNNs are trained on large datasets of labeled images to learn the features that differentiate between different types of objects. In the case of quality checking bricks, CNNs can be trained to recognize the surface texture, color, and defects in bricks that affect their quality.

Chapter-7

7.1 Conclusion

A CNN-based method was proposed for automatic drill core characterization from 2-D core tray images to produce less subjective results, provide a means to assess historical databases, and reduce the time and cost associated with routine RQD calculations and rock mass classification.

A CNN model was trained using a set of 6400 training images obtained from sandstone Score trays from different boreholes. The method was applied to automatically calculate the RQD by identifying intact core fragments and eliminating fractures, crushed cores, empty tray regions, and other objects present in the core trays. The RQD was calculated for each row in the core tray, resulting in an average error rate of less than 3% for new unseen sandstone tray images. In addition, the proposed method yielded accurate results when applied to limestone tray images, with an average error rate of approximately 3%.

The proposed approach promises an affordable, fast, and robust RQD calculation procedure that can be easily adopted and developed for further core logging automation. It offers remote and flexible core logging that does not require technical expertise. It relies on extracting geotechnical information from 2-D core tray images, which enables field assessment or reassessment using existing data sets without requiring access to old cores that may be damaged or lost. This method provides a new way to automatically analyze and quantify core data repositories, such as those available through Geoscience in Australia.

7.2 FUTURE SCOPE

The future scope for the project "quality checking of bricks using convolutional neural networks" includes expansion to other construction materials, integration with existing production lines, real-time quality monitoring, data analysis and prediction, and integration with robotics. This project has the potential to

significantly improve the brick manufacturing industry by increasing production rates, reducing waste, and improving the quality of products. As the technology evolves, it can be integrated with other industries, providing a solution for quality checking in a wide range of manufacturing processes

7.3 REFERENCE

- [1] Ali A, Shams Uddin SM, Ralescu AL (2015) Classification with class imbalance problem: a review. *Int J Adv Soft Compute Appl* 7:176–204
- [2] Alom MZ, Taha TM, Yakopcic C et al (2018) The history began from Alex Net: a comprehensive survey on deep learning approaches. *arXiv:1803.01164*
- [3] Alqahtani N, Alzubaidi F, Armstrong RT et al (2020) Machine learning for predicting properties of porous media from 2d X-ray images. *J Pet Sci Eng* 184:106514. <https://doi.org/10.1016/j.petrol.2019.106514>
- [4] Alzubaidi F, Mostaghimi P, Swietojanski P et al (2021) Automated lithology classification from drill core images using convolutional neural networks
- [5] Bao S, Chung ACS (2018) Multi-scale structured CNN with label consistency for brainMR image segmentation.
- [6] Barton N, Lien R, Lunde J (1974) Engineering classification of rock masses for the design of tunnel support. *Rock Mech* 6:189–236. <https://doi.org/10.1007/BF01239496>
- [7] Shawe-Taylor J, Zemel R, Bartlett P, Pereira F, Weinberger KQ (eds) *Advances in neural information processing systems*, vol 24. Curran Associates, Inc.,
- [8] Granada, Spain Bergstra J, Komer B, Eliasmith C et al (2015) Hyperopt: a Python library for model selection and hyperparameter optimization. *Comput Sci Discov* 8:0–24. <https://doi.org/10.1088/1749>
- [9] "Imbalanced deep learning by minority class incremental rectification" by Yan Y, Xu Y, Iosifidis A et al (2020): This research article proposes a method for imbalanced deep learning using minority class incremental rectification. It introduces a dynamic sampling strategy that focuses on the minority class

during training to improve the performance of deep learning models on imbalanced datasets.

- [10] "Deep residual networks for image classification with highly imbalanced datasets" by Sun H, Chen J, Cui S et al (2019): This research article presents a method using deep residual networks for image classification with highly imbalanced datasets. It proposes a novel deep residual network architecture that integrates oversampling and undersampling techniques to address the class imbalance problem.
- [11] "Focal loss for dense object detection" by Lin T-Y, Goyal P, Girshick R et al (2017): This research article introduces the focal loss, a loss function designed for dense object detection in computer vision tasks. It addresses the class imbalance problem by assigning higher weights to the misclassified samples of the minority class during training.
- [12] "EfficientNet: Rethinking model scaling for convolutional neural networks" by Tan M, Le QV (2019): This research article presents EfficientNet, a scalable convolutional neural network architecture that achieves state-of-the-art performance on image classification tasks with efficient use of computational resources. It provides a comprehensive analysis of model scaling, including the impact of class imbalance on model performance.
- [13] "Instance imbalance learning: an oversampling-based approach" by Chawla NV, Bowyer KW, Hall LO et al (2002): This research article introduces an oversampling-based approach called Instance Imbalance Learning (IIL) for addressing the class imbalance problem. It proposes a method that generates synthetic examples of the minority class using oversampling techniques to balance the class distribution in the training data.
- [14] "U-Net: Convolutional networks for biomedical image segmentation" by Ronneberger O, Fischer P, Brox T (2015): This research article presents the U-Net

architecture, a convolutional neural network designed for biomedical image segmentation tasks. It introduces a U-shaped network that combines convolutional and deconvolutional layers to achieve accurate segmentation of biomedical images, including applications such as brain MRI segmentation.

- [15] "Transfer learning for image classification with sparse label data" by Zhang Y, Zhou D (2018): This research article proposes a transfer learning approach for image classification tasks with sparse label data. It addresses the challenge of limited labeled data by leveraging knowledge learned from related tasks or domains to improve the classification performance on sparse label datasets.s