Garbage Classifier Through AlexNet Architecture

A Project Report Submitted to

Jawaharlal Nehru Technological University Anantapur

in partial fulfillment of the requirements for
the award of the degree of

BACHELOR OF TECHNOLOGY
IN
COMPUTER SCIENCE AND SYSTEMS ENGINEERING

Submitted by

Batch No: CSSE-25-32

SHAIK KATTUBADI SAMEERA	**21121A15A3**
PERUGU CHARAN RAJ	**22125A1511**
THIRUPATHI UJWAL	**21121A15B1**
PUCH SAI KUMAR REDDY	**22125A1513**

Under the Supervision of

Dr. Pydala Bhasha, *M. Tech, Ph.D*
Assistance Professor
Department of IT

Department of Computer Science and Systems Engineering
SREE VIDYANIKETHAN ENGINEERING COLLEGE
(AUTONOMOUS)
Sree Sainath Nagar, Tirupati – 517 102, A.P., INDIA

2024 -2025

Department of Computer Science and Systems Engineering

SREE VIDYANIKETHAN ENGINEERING COLLEGE

(AUTONOMOUS)

Sree Sainath Nagar, Tirupati – 517 102, A.P., INDIA

Certificate

This is to certify that, the Project work entitled

“**Garbage Classifier Through AlexNet Architecture**”

is the bonafide work done by

SHAIK KATTUBADI SAMEERA	**21121A15A3**
PERUGU CHARAN RAJ	**22125A1511**
THIRUPATHI UJWAL	**21121A15B1**
PUCHA SAI KUMAR REDDY	**22125A1513**

in the Department of **Computer Science and Systems Engineering**, **Sree Vidyanikethan Engineering College (Autonomous), Sree Sainath Nagar, Tirupati** and is submitted to **Jawaharlal Nehru Technological University Anantapur, Anantapuramu** for partial fulfillment of the requirements of the award of B.Tech degree in Information Technology during the academic year 2024-2025. This work has been carried out under my supervision. The results of this project work have not been submitted to any university for the award of any degree or diploma.

Supervisor:

Dr. Pydala Bhasha, *M Tech., Ph.D.*
Assistance Professor
Dept. of IT
Sree Vidyanikethan Engineering College
Sree Sainath Nagar, Tirupati – 517 102

Head of the Dept.:

Dr. K. Reddy Madhavi, *M.Tech., Ph.D.*
Professor & Head
Dept. of CSSE
Sree Vidyanikethan Engineering College
Sree Sainath Nagar, Tirupati – 517 102

INTERNAL EXAMINER **EXTERNAL EXAMINER**

DECLARATION

We hereby declare that the project report titled "**Garbage Classifier Through Alexnet Architecture"** is the genuine work carried out by us, in **B.Tech(Computer Science and Systems Engineering)** degree course of **Jawaharlal Nehru Technological University Anantapur, Ananthapuramu** and has not been submitted to any other college or University for the award of any degree by us. We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea / data / fact / source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Signature of the students

1.

2.

3.

4.

ACKNOWLEDGEMENT

We are extremely thankful to our beloved Chairman and founder **Dr. M. Mohan Babu** who took keen interest to provide us the infrastructural facilities for carrying out the project work.

I am extremely thankful to our beloved Chief executive officer **Sri Vishnu Manchu** of Sree Vidyanikethan Educational Institutions who took keen interest in providing better academic facilities in the institution.

We are highly indebted to **Dr. Y. Dileep Kumar**, Principal of Sree Vidyanikethan Engineering College for his valuable support and guidance in all academic matters.

We are very much obliged to **Dr. K. Reddy Madhavi**, Professor & Head, Department of CSSE, for providing us the guidance and encouragement in completion of this project.

We would like to express our indebtedness to the project coordinator, **Mr. P. Yogendra Prasad**, Assistant Professor, Department of CSSE for his valuable guidance during the course of project work.

We would like to express our deep sense of gratitude to **Dr. Pydala Bhasha**, Assistant Professor, Department of Information Technology, for the constant support and invaluable guidance provided for the successful completion of the project.

We are also thankful to all the faculty members of the IT&CSSE Department, who have cooperated in carrying out our project. We would like to thank our parents and friends who have extended their help and encouragement either directly or indirectly in completion of our project work.

Project Associates:

S.k Sameera

P. Charan Raj

T.Ujwal

P.Sai Kumar Reddy

ABSTRACT

High rises in the waste generated worldwide pose challenging problems to be dealt with formulating efficient waste management strategies. The basis of such strategies is the necessity of proper segregation of wastes into two categories: first one includes biodegradable materials that can decompose, and second one includes non- biodegradable materials, which are long lasting in the environment. It is not only important but also indispensable to clearly identify the different types of materials. Such clear identification is crucial in facilitating recycling efforts and fully alleviating the hassles that waste causes to our environment. Traditionally, the process of waste sorting relies heavily on manual processing or basic methods based on recognition of images. Traditional methods involve considerable variability in efficiency and accuracy of time, with a possible error in classification. Such incorrect classification can lead to contamination of materials that could otherwise be recycled. This decreases the general effectiveness of a recycling pprogram The project aims to introduce an advanced automated technique that operates on deep learning towards detecting and classifying the contents of dumpsters from waste items. It targets biodegradable against non-biodegradable material. It employs the use of architecture AlexNet, widely recognized for the effectiveness it has manifested in solving image classification challenges. Employing AlexNet provides avenues towards processing images that illustrate different materials to be categorized, developed through heavy training on the architecture. The dataset used in this project here comprises various images of waste materials differentiated between biodegradable and nonbiodegradable categories. To confirm the correctness of the AlexNet model, a very stringent comparison of its classification accuracy is done with that of several existing methods, hence providing an accurate account of its performance.

Keywords: waste management, image recognition techniques, deep learning, AlexNet architecture, biodegradable and non-biodegradable, environmental pollution.

TABLE OF CONTENTS

Title	Page No.
CERTIFICATE	i
DECLARATION	ii
ACKNOWLEDGEMENT	iii
ABSTRACT	iv
TABLE OF CONTENTS	v-vi
LIST OF FIGURES	vii
LIST OF TABLES	viii
ABBREVIATIONS	ix
NOTATIONS	x
CHAPTER 1 INTRODUCTION	11-12
1.1 Introduction of the Project	11
1.2 Motivation	12
1.3 Objectives	12
CHAPTER 2 LITERATURE SURVEY	13-36
CHAPTER 3 METHODOLOGY	37-50
3.1 Data Collection and Preprocessing	38
3.2 AlexNet Architecture Implementation	40
3.3 Model Training and Optimization	43
3.4 Model Evaluation and Performance Analysis	45
CHAPTER 4 SYSTEM DESIGN AND IMPLEMENTATION	51-57
4.1 System Requirements	
4.1.1 Hardware Requirements	52
4.1.2 Software Requirements	52
4.2 Design using UML Elements	53
4.2.1 Class Diagram	54
4.2.2 Use Case Diagram	55
4.2.3 Sequence Diagram	56
4.2.4 Activity Diagram	57
CHAPTER 5 RESULT	59-62

CHAPTER 6 APPLICATION	64
CHAPTER 7 CONCLUSION AND FUTURE WORK	66
REFERENCES	67-70
APPENDIX	70-81
CERTIFICATES OF CONFERENCE	82-84
COLLEGE VISION & MISSION	85
DEPARTMENT VISION & MISSION	86
PROGRAM EDUCATIONAL OBJECTIVES	87
PROGRAM SPECIFIC OUTCOMES	88
COURSE OUTCOMES	90

LIST OF FIGURES

Figure No.	Title	Page No.
Figure 3.1	AlexNet Architecture	41
Figure 4.1	Flow of Built Model	52
Figure 4.2	Class diagram	53
Figure 4.3	Use Case Diagram	54
Figure 4.4	Sequence diagram	55
Figure 4.5	Activity diagram	56
Figure 5.1	Accuracy curve	58
Figure 5.2	Loss Curve	59
Figure 5.3	Confusion Matrix	59
Figure 5.4	Non-bio degradable image	61
Figure 5.5	Bio-degradable image	62

ABBREVIATIONS

CNN	Convolution Neural Network
DL	Deep Learning
AI	Artificial Intelligence
RGB	Intelligent Red Green Blue(Color format of images
GPU	Graphics Processing Unit
ROI	Region of Interest
Q - Learning	A Reinforcement Learning algorithm
TPU	Transfer Processing Unit
IoT	Internet of Things
ReLU	Rectified Linear Unit(Used in Image Processing)

NOTATIONS

s	State of the system
a	Action taken by the agent
$Q(s, a)$	Q-value function, representing the expected reward for taking action 'a' in state 's’
$\gamma\ (gamma)$	Discount factor for future rewards
R	Reward received from the environment
$\in$ (epsilon)	Exploration-exploitation trade-off parameter in Q-learning
$\pi(pi)$	Policy function
$v(s)$	Value function for a given state
$\sum(sigma)$	Summation notation
$\alpha(alpha)$	Learning rate

CHAPTER 1
INTRODUCTION

1.1 Introduction

The world today is confronted with a mounting waste management crisis, fueled by increased urbanization, industrialization, and population growth. As cities keep growing, the volume of waste produced increases at an alarming rate. Conventional practices of waste segregation and management are becoming increasingly insufficient to cope with the sheer quantity and complexity of waste generated by urban . In this context, efficient waste classification has become an essential component of modern waste management systems, with the goal of improving recycling processes, minimizing landfill use, and reducing environmental harm.

Waste segregation, the separation of waste materials into the different streams is critical step in ensuring recyclables are well processed and non-recyclables are disposed of safely. Good waste segregation plays a significant role in avoiding contamination in recycling streams and enhancing the quality of the recovered material. The conventional technique of waste segregation is labor-intensive, error-prone, and typically inefficient. This has led to increasing interest in applying automation and emerging technologies, such as deep learning, to address these challenges. "Deep learning is a subset of machine learning and it proven to be very effective in the image classification and recognition tasks. Since they can process large amounts of the data and identify sophisticated patterns, deep learning models are most appropriate for the automation of the waste classification." By using deep neural networks (DNNs) to handle visual data, deep learning algorithms can automatically recognize and classify different types of trash based on the properties of shape, color, texture, and material. Deep learning technologies have created new possibilities to build highly scalable, efficient, and accurate trash classification systems. The likely benefits of using deep learning in waste sorting are many. Firstly, it can decreases labor costs and increases sorting speed and accuracy.

Moreover, the automated technologies are not tired and can sort enormous amounts of waste within a limited period of time, which is very helpful in business places or large industries. Additionally, deep learning models can also be trained to handle an immense range of waste products, even those that might prove difficult for human labor to identify or sort manually, such as small fragments of plastic, complex materials, or non-standard objects in the world. In municipal waste management, the ability to sort out recyclable materials of greatest importance at a fast and efficient rate. Recycling is a core component of the sustainable waste management practicals, which ensures the conservation of natural resources in the environment. Yet most recycling facilities still struggle separating the worthwhile recyclables from the non-recyclable materials. Tainted streams of waste may hamper the productivity of recycling, worthwhile material may find its way into landfills or incinerators. Such a system based on deep learning for classifying garbage can make it simpler by separating materials into their respective categories prior to recycling plants, ultimately enhancing recyclable material quality and recycling process efficiency through the utilization of waste management.

The other significant of the deep learning waste classification is it can improve, the through learning and adaptation. The ml models, after they trained on a exisring dataset of tagged models of waste, can be updated on their classification skill over time. When new forms of data of the waste items are generated or when waste disposal patterns alter, the system is retrainable to detect new forms of waste. This adaptability makes deep learning-based waste sorting systems highly scalable and resilient to variations in waste composition.

1.2 Motivation

The motivation behind the project is to decrease the waste in the environment other significant of the deep learning waste classification is it can improve, the through learning and adaptation. The ml models, after they trained on a exisring dataset of tagged models of waste, can be updated on their classification skill over time. When new forms of data of the waste items are generated or when waste disposal patterns alter, the system is retrainable to detect new forms of waste. This adaptability makes deep learning-based waste sorting systems highly scalable and resilient to variations in waste composition.Besides that, the systems can also be applied in various environments, from domestic recycling bins to industrial waste processing plants, thereby making them flexible and adaptable to various waste management environments

1.3 Objectives

Besides waste management, using deep learning, in garbage sorting also helps and pursuing higher environmental objectives.With higher levels of waste segregation and recycling, such systems can decrease the environmental hazard of landfills and incineration, of which have high environmental and health hazards. Landfills, for instance, produce extensive methane emissions, a highly reactive material that contribute to climate change. Incinerators have the additional burden of air pollution as well as venting harmful chemicals into the air. Proper segregation of waste would help in minimizing the dependency on such wastage disposal, thus making procedures for waste management more eco-friendly and less harmful to the environment.

CHAPTER-2

LITERATURE SURVEY

LITERATURE SURVEY

Zeyi Zhang et al.[1] proposed a system for automated garbage classification through deep learning to address the growing need for effective waste management systems. The adviced system employs CNNs in order to classify garbage into different categories in order to spur to on effective waste segregation systems. The authors emphasize using imaging recognition techniques and object detection systems in a quest for accurate classification outputs. Evaluation or the framework shows great promise under experimental conditions, indicating an important value for real world application in the context of smart cities and green waste disposal systems.

Limitations: While the system is promising, there are many problems with it. Deep learning techniques require a huge amount of computing resources, excluding deployment in systems with limited resources. [7]The performance is heavily depends on the quality and variation of training data, hence it becomes less efficient when presented with untrained or uncertain garbage types. In addition, practical application involves strong hardware infrastructure, such as cameras and processing units, which can be expensive for large-scale use. Future research should aim to optimize the model for lower-resource environments and solve scalability and adaptability issues to different types of waste.

Bowen Fu, Su Li et al.[2] conducted a study proposes an intelligent waste classification system based on deep learning and embedded Linux technology for solving environmental problems related to the management of garbage. The system consists of a combination of hardware and software modules which encompasses a CPU, an image sensor, sensors, and a touch panel as the input. The authors derive a GNet with transfer learning and an "enhanced MobileNetV3 architecture suitable for effective garbage classification". A easy-to-use GUI is implemented using Python and QT for smooth human-computer interaction. The model is capable of achieving a classification accuracy of 92.62% with a prediction efficiency of 0.63 sec on the 'Huawei Garbage Classification Challenge Cup dataset.'

Limitations: Though the results are promising; the system has limitations for real-world application. The use of high quality hardware components like Raspberry Pi and sensors adds cost that could inhibit scalability in resource-limited settings. The classification model's accuracy could be adversely impacted by untrained garbage types or changing environmental conditions. Finally, though the system works for that data set, it should be tested for uniformity of performance when applied to other data sets or real waste streams. Future work to develop the robustness of the model and to eliminate the reliance on costly hardware will be needed so the system can be more widely accepted.

Zhize Wu et al.[3] This study proposes an innovative benchmarking method called HGI-30 for knowing household trash images and addressing challenges presented by complex backgrounds, changing light conditions, different angles, and diverse shapes that complicate the classification of trash. The HGI-30 dataset comprises 18,000 images containing 30 categories of household trash resembling the conditions of living and representing real-world environments. [38]Moreover, the study conducts performance evaluations through state-of-the-art deep convolutional neural network (CNN) methods and provides baseline performance for garbage image classification. The dataset is intended to assist researchers in developing more robust and accurate recognition models for household trash, a vital step forward in advancing the field.

Limitations: “Although the HGI-30 dataset is a good source of garbage image recognition there are some limitations to be recognized.” The dataset might not encompass all the variations occurring in real-world garbage, like extreme weather scenarios or rare garbage objects, which might impact the models' generalizability trained on the dataset. Also, deep learning , particularly CNNs, demand enormous computational resources, which could potentially restrict their usability for researchers with limited resources. Moreover, the model performance on the HGI-30 dataset does not necessarily equate to real-world applicability without further validation and fine-tuning since real-world garbage sorting typically comprises more complicated and dynamic conditions.

Dan Zeng et al.[4] introduced a new approach to detecting garbage over large areas through the use of airborne hyperspectral imagery, overcoming the shortcomings of labor-intensive and time-consuming manual patrols. [20] It describes a Multi-Scale Convolutional Neural Network (MSCNN) for classifying hyperspectral image (HSI) data, in which pixels are classified to produce a binary garbage segmentation map. The data used for this reserch, the “Shandong Suburb Garbage dataset, is newly labeled and publicly released. The approach applies unsupervised region proposal generation methods, including Selective Search and Non-Maximum Suppression (NMS), to find garbage regions' locations and sizes. The experimental results demonstrate that the developed approach is robust in large-area trash detection and surpasses other HSI classification approaches on widely used public HSI datasets, like Indian Pines and Pavia University.”

Limitations: In spite of the encouraging outcomes, there are several limitations to the method. The use of hyperspectral data to detect rubbish may restrict its utility in areas without airborne hyperspectral sensors available, and therefore its use could be less viable in real-time or mass-scale implementations in certain parts of the world. The Shandong Suburb Garbage dataset, as valuable as it is, can potentially not encompass all garbage types or environmental contexts, which could hinder the generalizability of the model to other areas or garbage types. Moreover, computational resources needed for processing hyperspectral data and applying the MSCNN model might be extensive, preventing it from being used by certain users. In addition, it detects smaller or dispersed trash in extremely complex environments still needs to be fully assessed.

Cuiping Shi et al. [5] introduced limitations imposed by deep learning on garbage image classification owing to the unavailability of adequate training data are addressed. The authors concentrate on network architecture optimization to enhance performance on the popular TrashNet dataset. It is noted that short-circuit connections and deeper networks that are prevalent in deep learning perform poorly on the dataset. To help address this, the authors presented a novel method to augment the network where they introduce branches and apply extra layers to combine feature information. Their method efficiently leverages feature information without significant additional computation. They modified the basic architecture of the Xception network by using this method and achieved significant performance improvements. The new M-b Xception network achieved 94.34 percent classification accuracy on the TrashNet dataset, generally exceeding several techniques across many metrics and model is available for public use on GitHub.

Limitations: Despite [21]the M-b Xception network showing strong performance on the TrashNet dataset, this ability may be restricted to the characteristics of this dataset. The model may not have good generalisability to other garbage classification datasets, or garbage images from the real world because of the differences in image quality, lighting condition, and garbage type. The approach is also reliant on network expansion, which comes with more computational costs in training time and memory. "The need for large amounts of labelled training data continues to be a challenge, and while the approach shows an improvement in performance on TrashNet, the scalability of this approach to larger/more diverse datasets has yet to be fully investigated."

Wen Ma et al. [6] proposed a new algorithm called the "Lightweight Feature Fusion Single Shot Multibox Detector (LSSD) that aims to address inefficient and laborious manual garbage sorting processes. This technique aims to enhance the performance of the Single Shot Multibox Detector (SSD) algorithm by adding a novel feature fusion module". The algorithm boosts detection performance by combining features from various layers of different scales as well as creating a new feature pyramid structure with down sampling blocks. This feature pyramid is then passed on to multibox detectors for prediction. To deal with the unequal ratio of positive to negative samples, the authors use balanced cross entropic with Focal Loss which minimizes the loss weights for easy examples but prioritizes harder more beneficial examples. The authors also change to ResNet-101 as the backbone network replacing VGG16 which also improves detection performance. The Soft-NMS algorithm is also adapted to fine-tune the detection by removing unwanted bounding box predictions based on confidences scores, instead of simply dropping the bounding boxes entirely. The authors report experimental results that suggest the LSSD achieve better accuracy and time than other object detection methods.

Limitations: LSSD algorithm, while showing the aforementioned efficacy, might be utilized in real-world garbage detection scenarios to drastically reduce the potential performance of the algorithm.

Haruna Abdu et al.[7] explained the increasing focus on waste or garbage management, especially in both developed and developing nations, as part of intelligent and sustainable development efforts. Waste management entails various intricate processes, and in recent times, there has been a boom in applying ml methods to solve problems in this area and paper points out the rising interest in DL as a substitute computational method for solving waste management issues.

The authors note that while there have been some surveys on waste detection and classification, none have comprehensively explored the application of DL across various waste management problems. Furthermore, no study has systematically examined the datasets available "for waste detection and classification in different domains. To address this gap, the paper reviews various image classification and object detection models used in waste detection, providing an organized analysis of these techniques and compiling over twenty benchmarked trash datasets.The study also addresses the challenges faced by existing methods and discusses the future potential in the field. The goal is to provide researchers with a solid understanding of the state-of-the-art DL models and offer insight into potential research areas that are still unexplored. This survey aims to serve as a foundation for future research in waste detection and classification using deep learning, with the ultimate aim of improving waste management systems globally."

Haruna Abdu et al.[8] focused on the growing significance of waste or trash management, particularly in the context of sustainable development in both developed and developing countries. The waste segregation system consists of various complex processes, and deep learning (DL) has recently gained attention as an alternative computational technique for addressing waste management challenges. [31[The authors emphasize the increasing research interest in applying DL to waste management, noting that substantial research has been published in this field in recent years.

"The paper identifies a gap in existing surveys, as no comprehensive study has investigated the application of DL to waste management problems across various domains. Furthermore, there is a lack of research highlighting available datasets for waste detection and classification in different sectors. To address this gap, the authors review a range of image classification and object detection models, analyzing their applications in waste detection and classification. The paper provides a thorough and organized representation of the various techniques used, along with a compilation of over twenty benchmarked trash datasets

Cong Wang et al.[9] presented a "proof-of-concept municipal waste management system aimed at reducing the costs associated with waste classification, monitoring, and collection. "The proposed system leverages deep learning-based classifiers and cloud computing techniques to achieve high-accuracy waste classification at the initial stage of garbage collection. The system is designed to subdivide recyclable waste into six categories: plastic, glass, paper or cardboard, metal, fabric, and other recyclable waste, facilitating efficient waste disposal."

The authors employ investigating seven state-of-the-art CNNs and various data pre-processing methods. The classification accuracies for nine categories range from 91.9% to 94.6% in the validation set. Among the networks tested, MobileNetV3 stands out due to its "high classification accuracy (94.26%), small storage size (49.5 MB), and short running time (261.7 ms), making it an ideal choice for real-time applications in waste management."

Besides the deep learning model, the system devices to facilitate waste bins and the waste management center. The IoT devices have sensors that can measure the total quantity of waste generated within a particular region and monitor the operational status of every waste bin. The monitoring information facilitates adaptive scheduling of equipment deployment, maintenance, refuse collection, and vehicle routing schedules for the center of waste management, maximizing municipal waste management efficiency overall.

Sujan Poudel et al.[10] conducted a study on garbage classification is required to have a healthy environment. Segregation of waste by its type is one of the key elements of waste management. Due to advances in technology, especially hardware and artificial intelligence, waste segregation can be made automatic for better efficiency. "This paper investigates the application of convolutional neural networks

(CNNs) in classifying waste materials into seven types: cardboard, glass, metal, organic, paper, plastic, and trash.
The authors utilized pre-trained CNN models such as InceptionV3, InceptionResNetV2, Xception, VGG19, MobileNet, ResNet50, and DenseNet201 to determine the waste images.” The models underwent fine-tuning to enhance their effectiveness while implementing this task. The findings revealed that the accuracy was highest for InceptionV3 the models tested, and the VGG19 model had the lowest accuracy. Furthermore, the study classified waste into categories of Biodegradable (cardboard, organic, and paper) and Non-Biodegradable (trash, plastic, metal, and glass). Overall, the findings should foster confidence with respect to CNN-based transfer learning automated waste sorting, indicating a substantial potential role in the efficiency of waste management systems.
Betül Ay et al.[11] outlined the development of an image retrieval framework for e-commerce websites, which relies on the use of generative adversarial networks (GANs). The framework is able to retrieve images that are visually similar to a target product image (e.g. a shoe). The intent of the framework is to develop a content based recommendation system to rank and retrieve items that are visually similar to the queried item” .The authors compare the proposed GAN-based framework to the published state-of-the-art solutions and they use a benchmark dataset to validate the approach. Their results demonstrate the effectiveness of using a GAN to improve the accuracy and relevance of product recommendations, and indeed, it is a viable option for improving the shopping experience on e-commerce websites.This work contributes to an emerging body of research pertaining to visual similarity based recommendation systems, and it shows the potential of deep learning methods to improve product search and recommendation processes in an e-commerce setting.
Kyu Beom Lee et al. [12] presented a system for Object Detection and Tracking, which incorporates a deep-learning-model, Faster Regional Convolution Neural Network (Faster R-CNN), in order to automatically detect and keep track of unforeseen events in tunnels with poor CCTV viewing condition. It can detect serious events such as wrong-way driving (WWD), vehicles blocking passage, people exiting vehicles, and fires.” The ODTS handles video frames, detects objects, and tracks their movement by giving each detected object a unique ID. The system is trained using a tunnel event image dataset with high Average Precision (AP) values of 0.8479, 0.7161, and 0.9085 for car, people, and fire detection, respectively. The system was tested using four accident videos, with complete detection of all accidents within 10 seconds.One of the major strengths of the ODTS is that it can enhance detection performance automatically with an increase in the training dataset without any changes to the program code. The research proofs the capability of deep learning to improve safety and surveillance in tunnels by conducting real-time detection of major incidents.
Suchithra B et al.[13] reviewed and social media comments posing as users to mislead buyers and artificially increase product advertising has been the topic of this research paper. Authors suggest an automatic system to detect spurious reviews based on a modified random forest (ERF) algorithm The methodology employs sentiment analysis (SA) to analyze user responses in an effort to establish quality and authenticity of value that user reviews provide. “The authors compare the performance of an Extra Random Forest (ERF) with

other existing algorithm approaches including Decision Tree, Naive Bayes, SVM, and classical Random Forest. The results show that ERF outperforms the existing models in terms of performance ratio and runtime suggesting an effective method for detecting spurious reviews and commentary in an applied setting."

David J. Richter et al.[14]conducted a study on Double Deep Q-Learning (DDQN) application to fixed-wing aircraft attitude control introduces a novel application of reinforcement learning (RL) to the field of aviation. The authors are able to show convincingly that RL, in the form of DDQN, can be applied to solve intricate control problems conventionally needing expert insight and manual adjustment. Through the training of an agent to manipulate the attitude of the aircraft, the paper identifies the promise of deep reinforcement learning in automating and optimizing flight control systems. The application of the QPlane toolkit and simulators for training the agent provides a real-world approach to testing the system in a simulated environment prior to implementation. One of the greatest strengths of the paper is its discussion of how DRL algorithms that have already proven themselves in gaming and autonomous systems can be utilized in aviation, a domain requiring high levels of safety and performance. Nonetheless, as promising as the results appear, the paper would do well to also talk about some potential pitfalls, e.g., real-time adaptation for use in constantly changing environments, effects of sensor noise, and shifting from simulations to actual implementation. Despite these challenges, the research contributes to the growing body of work that shows DRL's potential to revolutionize autonomous systems, especially in industries where precision and reliability are paramount, such as aviation.

Wang Wang et al.[15] described a deep learning-based sentiment classification system for short texts with mixed languages using the Convolutional Neural Network (CNN) algorithm. As the datasets become more complex and the networks evolve, the paper investigates a range of deep learning classification algorithms such as multi-layer perceptrons, recurrent neural networks, and attention mechanisms." Authors emphasize the BL_CNN-based model construction and sentiment analysis, and they provide a precise mathematical model and training process for deep learning. The paper also illustrates how the CNN model handles datasets, gives a diagram of the neural network structure, and describes a regression model and emotion statistical clustering algorithm for prediction of sentiment. The system which has been proposed is for predicting sentiments from short text data even when the text is in mixed languages.

Limitations: Although the system has a good approach to sentiment analysis, it can be challenged by highly diverse or noisy data from multiple sources. The use of CNN for mixed-language text may fail to capture all sentiment nuances when compared with more advanced models such as transformer-based models. Also in practical applications where data is complex and diverse. Hybrid models or combining other deep learning methods might be investigated as future work to improve accuracy and generalizability across domains and languages.

Gozde Karatas et al.[16] discussed the application of deep learning methods to intrusion detection systems (IDSs), which are pivotal towards ensuring cybersecurity in today's networked world. As the number of networked devices grows and new forms of cybersecurity threats arise, IDSs play a vital role in detecting and reacting to novel types of attacks. “The article emphasizes the three primary elements of an IDS: data collection, feature selection/conversion, and the decision engine, with a special emphasis on how the decision engine can be improved through machine learning, particularly deep learning. The authors contend that deep learning presents a promising solution because it can process big data, decrease training times, and provide high accuracy. The article examines the literature on deep learning-based IDS techniques, comparing and contrasting different algorithms and their benefits in enhancing the effectiveness and efficiency of intrusion detection. The study provides a summary of the use of deep learning in intrusion detection systems, highlighting its potential to improve detection and response to cyber threats. Limitations: Despite promising advancements in deep learning for intrusion detection systems, the system must overcome challenges in terms of data quality and the complexity of attack patterns. Deep learning systems may require significant amounts of labelled training data, which may not be available, particularly for new and unknown attack modes. Furthermore, there are considerable resource obligations associated with the computational requirements of deep learning models for ultimate scalability in resource-poor environments.” Additionally, the model performance may be impacted by adversarial attacks or highly adaptive patterns of attack. Future work could focus on robust model performance, labelling methods and computational efficiency for educational applicability for real-world use.

Shuang Wu et al.[17] presented a overview of the application of deep learning algorithms for big data processing, a necessity today as we live in a world. The rapid development of technology and the resulting increase in the volume of data created in many fields has made it necessary to develop new ways to process data. Machine learning, which can learn from large datasets, presents possible solutions to this problem. ‘The paper describes various ML algorithms and methodologies such as supervised, unsupervised, and reinforcement learning, in the context of big data. The authors provide a synopsis of issues related to big data processing including volume, velocity and variety, and describe the weaknesses of current ML techniques with respect to these issues. The paper concludes by discussing ML applications in various fields including health, finance, and social media, citing that ML has improved decision-making and efficiency. “Overall this research provides an important contribution to understanding the relationship between deep learning and big data processing and establishes a nice base for academic area to follow. Limitations: Although the research provides a overview of methods for processing big data using ML, it does have some limitations.

Yann LeCun et al.[18] presented a thorough overview of the impact of deep learning on computer vision. The authors examine the advancements in deep neural networks (DNNs), convolutional neural networks (CNNs) that have revolutionized image classification, object detection, and segmentation problems.” The editors delve into the details regarding the theory behind deep learning and how it is superior to traditional methods, and it discusses the challenges of training deep networks, including overfitting challenges, computational resources, and large quantities of labeled data. The paper also delves into directions for future work in computer vision, including the incorporation of unsupervised learning, better network architectures, and more efficient training schemes. Through emphasis on major progress and challenges, the authors present a basis for further investigation and development in computer vision.

Limitations: Albeit the notable progress in computer vision with deep learning, the paper also recognises a few limitations. High computational cost with training deep networks is one big challenge that takes huge hardware requirements, e.g., GPUs and large datasets. Moreover, requirement of enormous labeled data can act as a limiting factor, particularly for those problems where labeling the data is time-consuming or costly. A second limitation is that of model interpretability since deep learning models, especially CNNs, are "black boxes" that do not reveal how they arrive at decisions. The authors propose that in future research efforts, the model's efficiency must be enhanced so that it needs less labeled data, and there is greater emphasis on the interpretability of deep learning models to render them more pragmatic and versatile for more varied applications.

Rajendra Akerkar et al.[19] presentsed an in-depth review of the use of machine learning (ML) in healthcare with emphasis on the different ML algorithms and their applications in medical fields. The authors discuss a series of techniques, such as supervised learning, unsupervised learning, reinforcement learning, and deep learning, for creating decision support systems in healthcare. These algorithms play important roles in applications such as disease diagnosis, treatment prediction, medical imaging, and healthcare management. The article points out the potential of ML to transform healthcare through identifying patterns in medical data and giving predictive information that can contribute to improved patient outcomes. The authors seek to minimize the gap in research on constructing effective decision support systems, which can aid healthcare professionals in making more accurate and timely decisions.”

Limitations: Although the paper emphasizes the potential uses of machine learning in medicine, it also identifies some challenges and limitations. One of the most prominent concerns is the quality and availability of medical data, since ML models need high-quality and large-sized datasets to train on. Data privacy and security concerns also exist, especially in handling sensitive patient information. Furthermore, although machine learning models have demonstrated high performance in disease prediction and treatment prognosis, interpretability of the models is still a major issue. “Most machine learning models, especially deep learning models, are typically black boxes”, and hence healthcare providers do not know why predictions were made. The study recommends that attention in the future should be drawn towards enhancing accessibility of data, model interpretability, and attending to ethical considerations to facilitate effective and safe utilisation of ML in healthcare.

Richard S. Sutton et al[20] provided an extensive introductory guide to reinforcement learning (RL), a new field of machine learning researching how agents may learn to make decisions with respect to their world in order to maximize the total sum of rewards. The authors introduce a number of substantial RL algorithms, including Q-learning, temporal difference learning, and policy gradient methods. They describe the use of RL in several roles, such as robots, video games, autonomous cars, and even decision making. The authors briefly touch on the challenges of scaling RL to real-world problems, including computational complexity, sample efficiency, and the usage of deep learning methods. The authors describe foundational RL, ultimately providing an overview of dynamic programming and temporal difference learning as overview. The authors discuss the history of RL algorithms, showing how some underlying principles have led to advancements in improving methods for tackling previously hard, real-world problems. The audience for this book is geared toward a high level of mathematic sophistication, providing detail on the theoretical advances that have been relevant and significant in defining RL as a field.

Limitations: The paper offers in-depth theoretical discussion of RL but because it's mostly grounded in mathematics, it may be difficult for readers with no math background to follow, Another limitation is its addressing obstacles to its use in real-world problems (e.g., computational complexity and data volume). The scalability of RL algorithms to more dynamic and unpredictable environments is still a large problem, and the paper could be improved with a few more concrete examples of RL's real-world applications. "Future work should focus on improving the computational efficiency of RL algorithms, improving their generalizability, and adapting them to various real-world problems."

Alireza Fathi et al.[21]created a comprehensive survey of deep learning methods in computer vision, with an emphasis on architectures including convolutional neural networks (CNNs), recurrent neural networks (RNNs), deep Boltzmann machines (DBMs), deep belief networks (DBNs), and stacked denoising encoders (SDAEs). The authors explain how these methods have transformed computer vision, and how the precision and performance of tasks like object detection, facial recognition, action and activity recognition, and human pose estimation have been greatly enhanced. The paper describes the historical progression, organization, benefits, and drawbacks of every architecture. It identifies how deep learning techniques have surpassed conventional machine learning algorithms in many computer vision tasks and how it has helped drive tremendous growth in the field. The review also gives a preliminary overview of how these deep learning models are being used in real-world computer vision applications.

Limitations: In spite of the vast improvements in computer vision brought about by deep learning, it is recognized in the main paper that deep learning algorithms tend to need extensive labeled data and a lot of computational resources to train with. Second, deep learning models have a limited generalization capability under some situations, particularly when the models trained are used on unseen, new data. The paper also identifies that deep learning models, though powerful, can be computationally costly and hard to interpret, which is challenging for practical use in real-world applications. Future work may address the efficiency of deep learning models, making them more feasible for resource-limited environments, and resolving the interpretability issues with these models.

Ross B. Girshick et al.[22] discussedthe history of object detection within the context of deep learning, with an emphasis on the progress achieved through enhanced object representation and the use of deep neural networks (DNNs). It categorizes object detection techniques into three broad categories: anchor-based, anchor-free, and transformer-based detectors, each providing varying means for detecting objects in images. The paper gives a comparison of different state-of-the-art object detection algorithms with respect to quality measures, speed/accuracy trade-offs, and training strategies. A detailed overview of prominent convolutional

neural networks (CNNs) for object detection is given, along with their advantages and disadvantages. Graphical representations illustrating the evolution of object detection approaches under deep learning are also provided by the authors, along with the future direction of research in this area.
Limitations and Challenges: The paper acknowledges that despite the great progress in object detection, the challenge of balancing accuracy and computational cost still exists. Different object detection models face different trade-offs, and finding optimal balance for real-time computational use is still a challenge. Some models also require large numbers of labeled images or data to train on, which could act as a restriction in some fields. Furthermore, the level of complexity of many derived deep learning models incurs computational costs and we can have difficulty interpreting the model, allowing us not to be optimal for use in real-world applications. Future work will be aimed at improving the efficiency, scalability and interpretability of the models, and address the problem of training with limited amounts of labeled data.
Siqi Sun et al. [23]cconducted a study on the enhancement of sequence-level networks, specifically "Long Short-Term Memory (LSTM) networks, to better handle structured input in machine reading systems. The authors propose a machine-reading simulator that incrementally reads text left to right and incorporates both shallow reasoning, memory, and attention. The model extends the standard LSTM model by replacing the standard memory cell with a memory network, which is more adaptive in its use of memory through neural attention in recurrence. This method enables the model to weakly induce relationships among tokens, which improve performance in tasks affiliated with language modeling, sentiment analysis, and natural language inference illustrated the ability of the LSTM-based model to be extended to different natural language processing (NLP) tasks, including question answering, sentiment analysis, and language translation. The paper also illustrates the ability to incorporate the model into an encoder-decoder architecture to support more advanced tasks. The experimental results conclude that the proposed model achieves or even surpasses the state-of-the-art techniques, especially for tasks involving sequential data handling and long-term memory storage."
Limitations and Challenges: The main difficulty of the presented LSTM-based machine reading system is to process rich, structured information. Although adaptive usage of the memory is facilitated through the memory network, issues such as model scalability for higher-order datasets or reasoning tasks are also possible. Besides, combining the LSTM with an encoder-decoder structure might improve the computational complexity of the model, thus making it less suitable for real-time operations. More research is required to optimize the memory network and enhance its generalization across a wide range of NLP tasks.
Tara N. Sainath et al.[24] worked on the issue of real-time keyword spotting system power consumption reduction, which is crucial in voice wake-up technology. The authors introduce a depthwise separable convolution neural network model and present two important techniques: approximate multiply and accumulate unit (AP-MAC) and streaming convolution reuse (SCR). These advancements lower the amount of computing power and memory space needed for KWS operations at high recognition accuracy. The model is trained using "Google's Speech Commands Dataset (GSCD) and lowers the usage of computing power by

37.7% to 42.6% over conventional Multiply and Accumulate (MAC) units," while yielding a comparable Word Error Rate (WER). In addition, the use of SCR technique enables reusing the result across several frames of audio, lowering activation storage by 94%. The combination of these methods yields a reduction of 98.5% to 98.7% in computing power and a decrease of 94% in memory storage for each audio frame.
The designed method is ideal for use in voice-controlled assistants and Internet of Things (IoT) devices where low power use and high recognition rates are essential. The article illustrates how deep convolutional neural networks (CNNs) can be leveraged in order to develop small-footprint models that execute properly on embedded systems, supporting real-time keyword recognition in speech.
Limitations and Challenges: Although the method reduces power consumption and memory usage considerably, it could be limited in scalability for larger or more complex datasets. Also, optimizing the model for various hardware platforms and maintaining real-time performance on various devices might be a challenge. Additional research may be necessary to extend the proposed methods to other KWS tasks and make the model more robust in noisy environments.
Wu Yanyan et al.[25] created a study on garbage classification has been an important issue in environmental protection and social livelihood. Conventional garbage classification methods are usually expensive and less precise. To solve these problems, this paper introduces a novel garbage image classification approach based on an enhanced VGG16 transfer learning model. The authors start by gathering and preparing a dataset of trash images, such as usual recyclable materials, toxic waste, kitchen trash, and other garbage. The VGG16 network architecture is fine-tuned by incorporating a convolutional layer with the SELU activation function after the initial five blocks of the model. The fully connected layer is also replaced with a global average pooling layer to prevent overfitting.
The model is trained by enhanced data augmentation-based dataset, and the model accuracy is tested. Comparative experiments illustrate that the enhanced model obtains an accuracy of 96.21% on the test set with a loss of 0.1203, and the training time has a dramatic decrease. "Experimental results indicate that the proposed approach exhibits significant advantages in garbage image classification. In addition, a garbage sorting apparatus is developed based on this approach, demonstrating its engineering applicati ons in practice."
Limitations and Challenges: Although the new method proves very accurate and efficient, there are likely challenges to scaling the model to accommodate more garbage categories or higher-order images. Further, fine-tuning the VGG16 model for certain types of refuse might be computationally intensive. Future work might involve optimizing the model for real-time use or investigating new architectures that might more significantly increase classification accuracy and decrease computational expenses.
Tara N. Sainath et al. [26] tackled the issue of real-time keyword spotting system power consumption reduction, which is crucial in voice wake-up technology. The authors introduce a depthwise separable convolution neural network model and present two important techniques: approximate multiply and accumulate unit (AP-MAC) and streaming convolution reuse (SCR). These advancements lower the amount

of computing power and memory space needed for KWS operations at high recognition accuracy. The model is trained using Google's Speech Commands Dataset (GSCD) and lowers the usage of computing power by 37.7% to 42.6% over conventional Multiply and Accumulate (MAC) units, while yielding a comparable Word Error Rate (WER). In addition, the use of SCR technique enables reusing the result across several frames of audio, lowering activation storage by 94%. The combination of these methods yields a reduction of 98.5% to 98.7% in computing power and a decrease of 94% in memory storage for each audio frame.

The designed method is ideal for use in voice-controlled assistants and Internet of Things (IoT) devices where low power use and high recognition rates are essential. The article illustrates how deep convolutional neural networks (CNNs) can be leveraged in order to develop small-footprint models that execute properly on embedded systems, supporting real-time keyword recognition in speech.

Limitations and Challenges: Although the method reduces power consumption and memory usage considerably, it could be limited in scalability for larger or more complex datasets. Also, optimizing the model for various hardware platforms and maintaining real-time performance on various devices might be a challenge. Additional research may be necessary to extend the proposed methods to other KWS tasks and make the model more robust in noisy environments.

Wu Yanya et al. [27] conducted a study ongarbage classification has been an important issue in environmental protection and social livelihood. Conventional garbage classification methods are usually expensive and less precise. To solve these problems, this paper introduces a novel garbage image classification approach based on an enhanced VGG16 transfer learning model. The authors start by gathering and preparing a dataset of trash images, such as usual recyclable materials, toxic waste, kitchen trash, and other garbage. The VGG16 network architecture is fine-tuned by incorporating a convolutional layer with the SELU activation function after the initial five blocks of the model. The fully connected layer is also replaced with a global average pooling layer to prevent overfitting.

The model is trained by enhanced data augmentation-based dataset, and the model accuracy is tested. Comparative experiments illustrate that the enhanced model obtains an accuracy of 96.21% on the test set with a loss of 0.1203, and the training time has a dramatic decrease. Experimental results indicate that the proposed approach exhibits significant advantages in garbage image classification. In addition, a garbage sorting apparatus is developed based on this approach, demonstrating its engineering applications in practice.

Limitations and Challenges: Although the new method proves very accurate and efficient, there are likely challenges to scaling the model to accommodate more garbage categories or higher-order images. Further, fine-tuning the VGG16 model for certain types of refuse might be computationally intensive. Future work might involve optimizing the model for real-time use or investigating new architectures that might more significantly increase classification accuracy and decrease computational expenses.

Haochang Hu et al. [28]studied on domestic waste pollution has also become a major source of environmental pollution in recent years, which has seriously damaged the global ecological environment and influenced people's living conditions. In response to the global trend towards environmental protection and sustainable development, many countries have started to promote garbage classification policies. However, conventional garbage classification technologies have their own shortcomings of heavy dependence on manual labor, low efficiency, and low sorting quality.

In order to solve these problems, deep learning methods have been more and more utilized in garbage classification. This paper offers an overview of the present garbage classification and detection technology based on deep learning. The authors summarize the research achievements in the area, pointing out the developments and advancements of different studies. The paper also summarizes the research work conducted by the authors in garbage classification.

Lastly, the paper presents the future issues and possible areas of further study in the area of garbage classification. These include enhancing the efficiency and accuracy of classification systems, creating more robust models, and dealing with the problems of real-time applications and scalability.

Junyian Wyng et al.[29] conducted a study on ever-developing economy, the amount of waste production has also been rising, making garbage pollution even worse and heavily influencing the environment. As the standard of living among people increases, the issue of waste management becomes more serious. In an attempt to tackle the problem, China has enforced a policy to categorize domestic wastes. Still, most of these programs are in their nascent stages and have yet to perform well in handling waste.

It's a main main headache in raw classification of the less containing of appropriate containers for a garbage collection system that will allow garbage to be sorted and recycled efficiently. The authors propose a new simple trash can for garbage classification. This structure has a garbage discharge port that is a specific shape and size so that the shape of the waste cannot be disposed of. In addition, the trash can has a relatively simple linkage system and a pressure plate that allows for compression of the trash when inputted into the bin. The use of this compression method provides an efficient storage capacity without the use of powered equipment. This is designed to improve the waste classification and sorting infrastructure by alleviating the problems with the existing systems. The authors state they could provide a true method for waste management at a relatively cheap alternative, particularly in developing countries that currently have poor waste classifying systems.

Li Cao et al. [30] leveraged waste resources, mitigate harms in society , and reduce the weightage of trash filtering by hand, work introduce a proceeding wastage recognizing, classification by shift the knowning. The authors implement the startingV4 model (initially trainedvisual data model for wastage classification. This method uses a pre-trained model and adapts it for classifying waste products by performing the task of transferring knowledge through fine-tuning the network.To maximize the models' performance, the authors first augment their dataset, so as to add to the diversity of their training data. The authors subsequently create a convolutional neural networks (CNN) based on the source model and then fine-tune their parameters based on the training results. The training results showed that the authors' model reached an achieved model of 99.4%, a testing performance of 92.4%. Finally, the authors tested their learnt model on real images gathered for the task of garbage classification. The achieved outputs reveal performs well, is able to correctly recognize commonly found garbage products.This approach becomes very useful in intelligent industrial waste classification, displaying its practicability and providing a good reference for future improvement in automated waste disposal.

Gengchen Yu et al.[31] conducted a review on garbage classification has become a more urgent problem. Yet, high classification accuracy and disposal are hindered by gaps in knowledge and awareness. Manual classification of garbage is usually inefficient, time-consuming, and subject to environmental interference. To solve these problems, the authors suggest an improved garbage classification approach based on an upgraded YOLOv7 object detection model.

The work proposes a new network structure named HorNet, which incorporates a recursive gated convolutional (gnconv) mechanism to enhance detection performance. The authors also add the C3HB module to the YOLOv7 model and optimize the pooling layer by replacing SPPFCSPC, resulting in enhanced detection accuracy. The model is trained on a self-built dataset for garbage classification tasks.

Acquired outputs reveal better way of impressively good compared to the native YOLOv7 model in accuracy. The new model records a mean average precision (mAP) of 99.25%, accuracy of 99.33%, and rate of recall as 98.03%. All these are compared to improvements in 1.50%, 3.99%, and 1.41% over the default YOLOv7 model. In general, achieved points demonstrate designed accuracy is high efficient on detecting garbage and classifying it, and it achieves significant improvements compared to the baseline model.

Shuijing Li et al.[32] worked on ecological conservation sustainable development become increasingly important, correct garbage classification is no longer optional. Yet public awareness of how to classify garbage is still lacking, and proper waste sorting becomes problematic. In an effort to remedy this, the authors have designed Mark Scoring in terms of garbage classification MSGC (region-based convolutional neural networks) with the goal of helping people to accurately classify garbage.

The study involves collecting new data from the Beijing municipal domestic waste classification standards. A method is performed for recognize and change other means of garbage with Mask Scoring RCNN. The objective is to develop a method that not only assists in effective waste classification but also saves time in the process of sorting waste and recyclables. The Acquired outputs show on the exhibited model exhibits a demonstrably more level of inline method on waste classification. Additionally, the best Mask Scoring RCNN methodology demonstrates better segmentation and recognition scores overall, and particularly in cluttered images. The system could effectively assist users in classification of the garbage to make the process more accurate and more convenient. Furthermore, the practicability of this technique indicates it could serve as a useful catalyst to prompt improved waste management practices overall.

Tian Li et al.[33] addressed the problem of creating a garbage classification model with high accuracy, a small model size, and real-time performance. The authors propose a wastage sharing output by an enhanced shutternetV3 light network. The enhancement the authors suggested enhances a coming exception capability as the CNN optimizing basic building block of ShufflenetV2 and incorporating the attention mechanism module, with particular reference to CBAM (Convolutional Block Attention Module). Ultimately, the optimization seeks to reduce minimum for the units used on one fingers on connection design, which simplifies any potential dimension reduction on model's parameters and complexity.This method sidesteps the problem of networks being too deep, where unnecessary information can be extracted, diminishing the accuracy. Moreover, the ReLU activation function is substituted with LeakyReLU to further enrich the feature information obtained by the network. Label smoothing applied to the loss function also counteracts class imbalance, minimizing its adverse effect on model performance.

Experimental findings indicate that the algorithm attains an accuracy of 81.26% in a self-built dataset. The number of the model's parameters is around 0.917 M, the computational burden is around 92.75 Mlops, and 182.32 MAdd. In contrast to ResNet101, the algorithm performs better in terms of accuracy but consumes only 1/44 of the parameters. This shows the potential of the enhanced ShufflenetV2 for deployment in resource-limited devices, including mobile terminals, providing a realistic solution for real-time garbage identification and classification.

Xiaodong Zhang et al. [34] solved the increasing issue of domestic trash sorting in China, which now stands as the world's leading waste generator. Due to the problem, China has identified principal cities such as Beijing, Shanghai, and Chongqing to implement waste classification. Despite that, garbage classification continues to meet major obstacles like a lack of public awareness and ambiguous classification requirements. To solve these problems, the authors introduce a classification recognition approach hands of YOLOv5, on the inter national finding of an item model. This way begins by labeling different kinds of garbage images and then training a classification model after cropping images. The trained model is then used for recognition on the YOLOv5 algorithm. Experiments testing the YOLOv5-based algorithm have demonstrated high accuracy, light-weight architecture, and robust performance, making it suitable for identification of resident's household waste. The model loads the data for residents to enter garbage, and can sort it out automatically, which can avoid the overhead of sorting it manually, and help improve waste disposal efficiency. This approach shows possible large-scale applicability in sorting of waste at the domestic level.

Alex Krizhevsky et al. [35] presented a special deep convolutional neural network (CNN) that was trained to classify 1.2 million high-resolution ImageNet LSVRC-2010 images. The CNN model demonstrated remarkable performance, achieving 37.5% and 17.0% top-1 and top-5 error rates, as of the, significantly goodest compared to all art of the nation methods. Architecture of a network is composed for five convolutional layers, pooling the maximum layees of , and four partially interpretted connectors, with a last 99 route Hardmax layer. This model has 60 million parameters and 50,000 neurons. The authors using a non-saturating neuron activation, combined with an incredibly efficient GPU-based convolution operation. To alleviate overfitting, the authors proposed a very successful regularization process known as dropout, which is affective at myness connecting in high attached things. This CNN model, popularly referred to as AlexNet, was then trained 16.4%, outperforming the second place entry (26.2).

Abien Fred M. Agarap et al.[36] described using Linear Units of rectifying (ReLU) as part of these hardest networks (DNNs). has been employed in the convnet literature as an activation function and is often not used for classification, while often in DNN literature as a function. This paper investigates other classification functions that contribute to existing literature that conflicts the standard methodology. In this method, the neural network activation from the second to last layer is computed, re-scaled by weight parameters to obtain raw scores, and then a threshold function (ReLU) was applied to the raw scores. The output class predictions are derived from the application of the argmax function to the thresholded scores. The paper seeks to offer insights into how ReLU can be utilized as a classification function, growing the knowledge base of deep learning architectures.

Limitations: Although the current research proposes a new way of performing classification in deep learning, it doesn't make an in-depth comparison between the performance of ReLU-based classification function and standard Softmax on numerous real-world applications. The merits and limitations of applying ReLU as a classification function aren't deeply analyzed for real-world performance, and additional empirical work needs to confirm its suitability for use in many scenarios. Moreover, the theoretical basis and interpretability of applying ReLU to classify in deep neural networks may be further researched.

Ibrahim A et al.[37] introduced a better activation function, Rectified Linear of Adaptive Unit (AReLU), to address the "problem dying" in deep models, especially (CNNs). The Rectified Linear Unit (ReLU) is popular because it is simple and performs better than previous activation functions such as tanh and sigmoid since it does not suffer from the vanishing gradient problem and has lower computational complexity. But ReLU has the dying problem, where neurons get inactive and do not learn during training. To counter this, the author suggests an extension of Leaky ReLU (LReLU), which dynamically modifies the loss function values during training with an increase in epochs. The model was experimented on the MNIST dataset for 10 epochs, with a 1.2% reduction in misclassification rate over conventional approaches. The method suggested is characterized by simplicity, low computational complexity, and the lack of hyperparameters, rendering it an effective solution to the dying problem of deep learning models.

Limitations: Although the proposed AReLU activation function performs well in decreasing the misclassification rate and resolving the dying problem, its generalization to more intricate and larger data sets other than MNIST is not well evaluated. Furthermore, the paper lacks comparison with other sophisticated variations of ReLU on different deep learning models, like CNNs for photo recognition or recurrent neurals are for sequence data. More experiments must be conducted to assess its generalized applications.

.Zheng Qiume et al.[38] proposed a novel activation function, slowly using of a Linear Unit (SELU), which is expected to solve frequent problems in deep convolutional neural networks, i.e., gradient vanishing, neuron death, and output bias. FELU is focused on speeding up exponential straight units and shortening network taking time while inheriting the merits of Linear Unit of Rectified (ReLU) and Exponential Linear Unit

(ELU). The authors tested experiments on CIFAR-10, CIFAR-100, and GTSRB datasets, comparing FELU with common activation functions like ReLU, ELU, SLU, MPELU, and TReLU. Experiments show that FELU not only increases classification accuracy but also enhances the calculation speed of exponential operations, resulting in improved network speed. FELU was also found to have enhanced noise robustness, which further led to increased classification accuracy.

Limitations: Although the FELU activation function has been shown to improve both its speed and accuracy of performance in the paper, its effectiveness has not been evaluated on more extensive datasets or on real-world implementations, where high-level data such as large amounts of data or complex network architecture might affect its performance. Moreover, the comparison with only five traditional activation functions narrows the scope of the evaluation, as new or hybrid activation functions might offer a more broad understanding of FELU's benefits. More extensive testing on various datasets and network topologies is required to confirm its generalizability.

Wenlino Sdang et al.[39] studied the characteristics of CNNs & presents a new gotup neurals named concatenated ReLU (CReLU). The researchers noticed that filters in lower layers of CNNs tend to come in pairs with opposite phases, which motivated the creation of CReLU. The proposed activation function concatenates the ReLU outputs with its negative pair, which enables improved representation and enhanced performance. The work incorporates CReLU into a variety of art of a nation CNN frameworks and assesses how recognition performance is affected and ImageNet pre processing sets. It demonstrates that CReLU improves recognition accuracy with fewer trainable parameters, illustrating how an improved understanding of CNN properties can translate into profound improvement with slight alteration of the architecture.

Limitations: Although the CReLU activation function offers performance gains, the paper does not touch on potential issues with extending this approach to more sophisticated or real-time scenarios where computational efficiency and model deployment may become increasingly important. The comparison of CReLU to just a few other existing activation functions may fail to highlight its strengths compared to more recent or hybrid activation functions. Additional testing on varied real-world data sets and more sophisticated network topologies would be required to confirm its universal applicability**.**

Rekiya Yemashita et al. [40]provided an overview of convolutional neural networks (CNNs) and their application in the field of radiology. CNNs have become a dominant tool in computer vision is to enhance the working of radiologists and patient care by incorporating CNNs into imaging through backpropagation. The authors explore how CNNs can be applied to various radiological tasks, including medical image analysis, and discuss the challenges faced when using CNNs in radiology, particularly issues like small datasets and overfitting. The article also outlines techniques to address these challenges and emphasizes the importance of understanding both the advantages and limitations of CNNs in diagnostic radiology. The aim is to enhance the working of radiologists and patient care by incorporating CNNs into imaging workflows.

Limitations: Although CNNs have a lot of potential in radiology, the paper indicates the limitation of small data sizes, which can cause overfitting and decreased model generalizability. While methods used to circumvent such limitations are mentioned, the paper does not make extensive comparisons among these methods and their efficiency across different clinical environments. In addition, although CNNs have the potential to improve diagnostic performance, their implementation in clinical radiology practice can be discouraged by issues of data privacy, the requirement for large annotated datasets, and incorporation of CNN-based systems into established radiology workflow. More research and development are required to overcome these practical hurdles and facilitate safe and effective deployment of CNNs in clinical settings.

Guifang Lin et al. [41] discussed a better variant of the (ReLU) activation function that aims for improve the performance of (CNNs) in applications like target detection and image classifying. The authors introduce a piecewise activation function, named SignReLU, which solves problems like gradient vanishing and slow convergence of the conventional CNNs. The model further includes local response normalization as well as maximum stacking layers. The new activation function is applied on the CIFAR-10 benchmark using TensorFlow, and experiments result in SignReLU clearly improving the convergence rate and image recognition accuracy versus standard ReLU. The enhanced activation function solves the issue of the gradient vanishing and improves the performance of the CNN overall.

Limitations: The research, although promising, concentrates on the CIFAR-10 dataset, which might not be representative of the difficulty of implementing the proposed activation function for higher-complexity or real-world datasets. The paper also does not include an exhaustive comparison with other state-of-the-art activation functions regarding computational efficiency and scalability. The overall generalizability of the proposed approach to other fields, outside of image recognition, is left as an area to be further explored. Also, the paper does not address the possible effect of the suggested activation function on the model's interpretability, which is an important consideration in a lot of real-world applications.

Rikika Yemashita et al.[42]worked on summary of convolutional neural networks (CNNs) and their uses in radiology, an area that has experienced growing interest in taking advantage of deep learning to analyze medical images. CNNs are created to learn spatial hierarchies of features automatically using backpropagation, which is especially appropriate for activity like medical image diagnosis and classification. The article talks about the problems in using CNNs for radiological applications, including the small size of annotated medical datasets and overfitting. Methods for solving these issues, like data augmentation and transfer learning, are also discussed. The authors note the promise of CNNs in enhancing the capability of radiologists and patient care, but they also point to the limitations and ethical implications in using these models in clinical environments.

Limitations: The paper points out major challenges in using CNNs in radiology, especially the problem of small datasets and overfitting. Although methods to counter these challenges are mentioned, the paper does not give detailed analysis or comparisons of these methods for various radiological tasks. In addition, the integration of CNN-based systems in existing clinical practice is still a main challenge, and the paper does not approach completely the technical and regulatory hurdles for large-scale adaptation in medical practice. The importance of transparency and interpretability of CNN models is also a key issue that is not thoroughly investigated in this review.

Ya Yinge et al. [43]introduced a novel activation function called Rectified Exponential Unit (REU) for Convolutional Neural Networks (CNNs). REU is inspired by two existing activation functions: The study examines the benefits of Swish and Exponential Linear Unit (ELU) activation functions, emphasizing their multiplication-based structure and flexible exponent. The authors present the Parametric Rectified Exponential Unit (PREU), which is intended to increase the expressive power of the Rectified Exponential Unit (REU) and further improve these advantages. Using benchmark datasets such as Fashion-MNIST, CIFAR-10, CIFAR-100, and Tiny ImageNet, they assess REU and PREU across popular CNN architectures, such as LeNet-5, Network in Network, and ResNet. According to the findings, both REU and PREU outperform conventional activation functions like ReLU in terms of classification accuracy. Particularly, when combined with ResNet, REU lowers relative error by 7.74% on CIFAR-10 and 6.08% on CIFAR-100, while PREU achieves even higher improvements of 9.24% and 9.32%, respectively. Furthermore, the study shows that various Limitations: While the proposed REU and PREU activation functions show promising improvements in performance across multiple CNN architectures and datasets, the paper does not provide an in-depth analysis of the computational complexity or the training time required for these new functions compared to traditional ones like ReLU. Additionally, the results are based on relatively standard datasets, and the generalizability of REU and PREU to more complex, real-world datasets or domains beyond image classification remains to be fully explored. Furthermore, the paper does not address the potential impact of REU and PREU on model interpretability, which is an important consideration for deployment in critical applications such as medical imaging or autonomous driving.

Netanel Friedenberg et al. [43] discussed the completion of casual toric varieties valuation of an over rank rings, focusing on equivariant open embeddings in normal toric varieties. It demonstrates that these varieties can be completed after a base change by a finite extension of valuation rings. If the value group is discrete or divisible, no base change is required. The paper also highlights that existing methods fail to produce such normal equivariant completions, and the author's approach provides a combinatorial solution. completions, the paper introduces a combinatorial analog of Noetherian reduction, which is expected to have independent mathematical interest.

Limitations: While the approach offers a combinatorial perspective on the completion of toric varieties, it is highly specialized and requires a deep understanding of algebraic geometry and combinatorics. The results may not be directly applicable to broader areas without further exploration of how these methods can be generalized. Additionally, the need for a finite extension of valuation rings may limit practical applications in certain contexts. Future research would be to extend these findings to broader contexts or trying to make the construction of admissible completions easier.

Sree Kruthikha D et al.[44] discussed the application of deep learning to automate garbage classification with the goal of enhancing waste sorting and recycling. The research targets the classification of waste into six groups: glass, paper, cloth, trash, cardboard, and plastic, based on (CNNs). A number of deepest knowning points, such as MobileNet, NASNet, LeNet, Inception, and DenseNet, were good at a datasheet that was curated to know intricate features from images of garbage. Among the models, NASNet indicated the greatest accuracy, hence making it most ideal for application in real-world situations, particularly within resource-restricted settings such as mobile or edge devices. It indicates that automated garbage sorting could enhance recycling effectiveness, decrease the amount of sorting required by human beings, and promote waste management sustainably, showing how deep learning could potentially revolutionize waste management systems.

Limitations: The system is promising but demanding in terms of computational power, particularly to train deep learning models, which might not be possible in all settings. High accuracy depends significantly , and the model can suffer with untrained or unclear waste types. Real-world implementation of the system would also call for solid infrastructure, such as cameras and processing power, which would be expensive. Future efforts would be dedicated to improving the models for low-resource environments as well as mitigating scalability and adaptability to various waste types.

Sujan Poudel et al [45] addressed the challenge of classifying waste, in particular Municipal Solid Waste (MSW), into categories of bio, plastic, glass, metal, and paper. The research discusses current deep learning methods of waste classification and suggests an architecture derived from EfficientNet-B0, a compound-scaling model that is pre-trained on ImageNet. The graphics of the model is optimized using shifting learning, which is customized to particular demographic areas for more effective classification. The study proves that the EfficientNet-B0 model, when fine-tuned for regional litter images, has similar accuracy to EfficientNet-B3 but with much fewer parameters, resulting in a 4X reduction in FLOPS. This method provides a more efficient and region-specific solution for waste classification.

Limitations: While promising, this method relies on the availability of region-specific litter images for fine-tuning which may not be the case everywhere. Furthermore, the effectiveness of the model is also contingent on the quality and diversity of dataset images, the model might be unable to generalize to waste types that are not captured in the classes it requests. Moreover, while the model produced is highly computationally efficient, it will still require reasonable computational resources for training and deployment. Future research should seek to expand the dataset to cover a wider area of wastes, as well as increase the model's adaptability to various locations and situations.**Retu Chahan et al.[46]** addressed the drawbacks of traditional waste monitoring techniques andemphasizes the significance of utilizing smart technology to improve waste management efficiency. In order to reduce human intervention and maximize sorting productivity, the authors suggest a deep learning-based method that uses convolutional neural networks (CNNs) to automate waste classification. In order to create a more economical and effective system, the study investigates different waste disposal methods and management technologies. The suggested CNN model outperformed other models, including AlexNet, VGG16, and ResNet34, demonstrating its potential for sustainable waste management and smart cities. However, the model's efficacy is limited by the caliber and variety of training data, even with its encouraging outcomes. Unfamiliar or unclassified waste categories may have an impact on its performance.

Moreover, the level of computational power required for training and for deploying the CNN model in real-time may be a limiting factor, particularly in resource-poor environments. Lastly, the performance of the model relies on obtaining high-quality images, which is not always feasible during real-world applications.

J. Chandrika et al.[47]examined the cause of learning techniques to classify garbage with the goal of improving efficiency in the recycling chain. The authors investigate different approaches to classify waste from images as: (SVM) with histogram of oriented gradients (HOG) features; simple (CNN) and CNNs with residual blocks. The authors conclude that simple CNNs, with or without residual blocks, perform well for garbage classification, which shows the potential of deep learning models to address recycling waste sorting problems. These methods can also significantly decrease the time and increase the accuracy of the recycling process while saving money at the same time. Limitations: Despite promising results, deep learning models, particularly CNNs, are difficult to generalize to different waste types; the performance of the models relies heavily on the variability and quality of the training data. This is because the classification model will not be able to classify waste types that were not originally included in the training set, making the system less robust overall. Furthermore, the computational resources to train the models, and subsequently deploy the CNN models, present challenges for real life applications and resource constrained situations. Future work may focus on improving model generalization by further investigating the computational costs of model deployment and operations to provide more scalable solutions.

Sivaranjani S et al.[48] examined several waste segregation methods that incorporate cutting-edge technologies, such as artificial intelligence (AI), deep learning (DL), machine learning (ML), Arduino, and the Internet of Things (IoT). The study examines five different strategies meant to improve waste management systems' effectiveness. The IoT system employs sensors and cameras to detect various types of wastes, send data, and initiate sorting systems such as robotic arms for remote monitoring and automatic sorting of waste. A sophisticated trash surveillance system using Arduino Uno identifies the types of garbage, tracks levels in bins, and triggers full bin alerts. Deep learning identifies waste images via convolutional neural networks (CNNs) at 93.3% accuracy for waste alert management. Machine learning-based models for sorting non-biodegradable plastics attain accuracy levels above 80%. The paper concludes in the Waste Segregation and Disposal (WSD) model, which uses IoT, Arduino, ML, DL, and AI technologies together to give real-time information and effective sorting by minimizing manual efforts and maximizing recycling practices.

Limitations: As attractive as are the features of latest technologies, deploying these waste segregation systems has limitations in the form of needing enormous infrastructure and resources. As impressive as are the accuracy of waste classification models, it still largely and the ambient conditions under which the system is deployed. Further, the mass installation of these systems could be challenged by the high cost of devices, including sensors and robotic arms, and technology integration complexity. Future research may target lowering the cost of the systems, enhancing model flexibility across different types of waste, and ensuring scalability to operate in varying environments

Qingqiang Chen et al.[49] discussed the surging requirement for effective garbage categorization processes with conventional dumping processes becoming obsolete based on the accelerating amount of refuse. An advanced YOLOV4 network model is developed to identify the garbage in terms of 15 objects grouped in 3 classes. The system is meant to eliminate constraints related to prior garbage classification software based on one-class recognition, insufficient types of objects, and less accurate classification. The results show that the advanced YOLOV4 model achieves an average accurate of 65% at a rate of 92 FPS. YOLOV4 model is suitable for waste classification tasks, and is most useful when run on embedded devices. Limitations: While the proposed solution outperforms existing algorithms, 64% accuracy would not be sufficient for many applications that require more accuracy. The on YOLOV4 could also further restrict the system from handling more complex or diverse types of waste. Also, the model could be negatively affected by the training data's quality or diversity, and the system may struggle to operate in real-world scenarios where waste is richer or more ambiguous. Future work would benefit from improved accuracy, tuning the model to run on real-time on embedded devices, and developing methods to process a richer set of waste categories.

Rushnan Faria et al.[50] examined the growing issue of waste accumulation, especially in cities where improperly managed waste poses major environmental hazards. Effective segregation is essential because a sizable amount of waste is recyclable. Manual sorting, however, is costly and time-consuming. In order to tackle this issue, the authors suggest an automated classification system that divides waste into four categories: glass, metal, plastic, and organic waste. They present the OrgalidWaste dataset, which consists of approximately 5,600 photos taken from different waste datasets. This dataset was used to assess several convolutional neural network (CNN) models, such as a 3-layer CNN, VGG16, VGG19, Inception-V3, and ResNet50. With an accuracy of 88.42%, VGG16 outperformed the others. According to the authors, by simplifying classification and increasing recycling effectiveness, this automated method can greatly improve waste management. Limitations: While the proposed method demonstrates a high accuracy of 88.42%, this may not be sufficient for real-world applications that require a higher degree of accuracy of categorization in the field in particular with many waste types. The performance of the model have the potential to be affected by the quality of the training data and diversity of the inputs, as the system may not perform well on new or unknown waste types. Additionally, some of the structure used in the model could limit the model's scalability or flexibility in more complicated scenarios. Future research must focus on increasing the model's generalizability, increasing the accuracy level, and real-world testing for further confirmation

CHAPTER-3
METHODOLOGY

METHODOLOGY

The design of the garbage classifier based on the AlexNet architecture is systematic in nature and ensures effective classification of various types of wastes. The approach has several stages such as data acquisition, preprocessing, implementation of model architecture, training, testing, and deployment. All these modules are essential in order to have a high-accuracy classification model that can identify various types of garbage for efficient waste management and recycling.

3.1. Data Collection and Preprocessing

The initial step towards creating a garbage classification system is obtaining a diverse dataset of images that comprise multiple kinds of waste materials. The dataset is compiled from various sources, such as publicly available datasets like TrashNet, Kaggle Garbage Classification Dataset, real-world images of urban and rural locations, and images taken from smart trash bins and IoT-enabled surveillance systems. Moreover, photos are collected from garbage dumps and recycling plants to include a large variety of waste materials like plastic, paper, glass, metal, organic refuse, and electronic refuse. This way, the model is capable of generalizing well across the different types of wastes.

After the dataset is gathered, it is preprocessed to enhance the quality and uniformity of the images. The processing stage involves cleaning of images and noise removal in order to remove blurred or low-resolution images. Image resizing and normalization are then carried out to have all images in the same dimensions (227×227 pixels) as the AlexNet architecture demands. In addition, data augmentation methods like rotation, flipping, brightness change, and injection of Gaussian noise are employed to make the model more robust and avoid overfitting. Last but not least, the dataset is divided into training (80%), validation (10%), and test (10%) sets to allow for appropriate evaluation and fine-tuning of the model.

1. Import Required Libraries:
   - os, tqdm, matplotlib, pandas, numpy, cv2, seaborn, random
   - Keras and TensorFlow for deep learning
   - sklearn for train-test split
   - glob for file handling

2. Extract Dataset:
   - Use `unrar` to extract the dataset from the given file path.

3. Set Dataset Path:

   - Define the path where the dataset is stored.

4. Read Folder Names:

   - Iterate through the folders in the dataset directory.

   - Print the folder names and the number of images in each.

5. Collect Image File Paths:

   - Traverse through all subdirectories.

   - Extract image file paths, ignoring database (`.db`) files.

   - Store the valid image paths in a list

6. Extract Class Labels:

   - Extract class labels from the folder structure.

   - Store the corresponding class label for each image.

7. Shuffle Data:

   - Pair image paths with class labels.

   - Shuffle the data randomly to avoid biased training.

   - Split shuffled data back into separate lists.

8. Create a Pandas DataFrame:

   - Store `image_path` and `Class_label` in a structured DataFrame.

Explanation of the Code:

1. **Importing Required Libraries:**
   - o Libraries like **os**, **cv2**, and **glob** are used for file handling.
   - o **Matplotlib & seaborn** help in data visualization.
   - o **Keras & TensorFlow** are used for deep learning.
   - o **sklearn** is used for dataset splitting.
2. **Extracting Dataset:**
   - o The dataset is extracted from a **RAR file** using the unrar command.
   - o This ensures that the image dataset is available for processing.
3. **Reading and Displaying Folder Names:**
   - o The script iterates over dataset folders to print **folder names** and **image counts**.
   - o Helps in understanding dataset structure.
4. **Collecting Image File Paths:**
   - o The script walks through the dataset directory to fetch all image file paths.
   - o It **ignores .db files** to prevent unwanted processing.

5. **Extracting Class Labels:**
    - Since images are stored in class-based folders (e.g., "Tomato", "Apple"),
        - Class labels are extracted from the **folder name**.
6. **Shuffling the Dataset:**
    - Shuffling ensures that training and testing are not biased toward any class.
    - The script **zips image paths & labels**, shuffles them, and then **unzips** them back.
7. **Creating a Pandas DataFrame:**
    - The structured DataFrame stores **image paths** and **class labels**.
    - This DataFrame can be used for **data loading, model training, and analysis**.

3.2. AlexNet Architecture Implementation

AlexNet is a deep convolutional neural network (CNN) that has been specifically created for large-scale image classification. It has five convolution layers, three fully connected layers, dropout regularization, and ReLU activation functions to provide effective feature extraction and learning. The input layer of AlexNet accepts RGB images of 227×227×3 pixels. The first convolutional layer uses 96 filters of 11×11 size with a stride of 4 and ReLU activation for the introduction of non-linearity. The second convolutional layer uses 256 filters of 5×5 size, with local response normalization (LRN) added to enhance feature discrimination. The third, fourth, and fifth convolutional layers use 384, 384, and 256 filters, respectively, with 3×3 kernel sizes for deeper feature extraction.

Following convolutional processing, the features are fed into three fully connected layers. The first and second fully connected layers have 4096 neurons, each preceded by ReLU activation to boost learning ability. The last layer is the output layer with a softmax classifier that outputs probabilities of various garbage categories, allowing multi-class classification. Overfitting is prevented using dropout regularization, which disables neurons randomly during training. Batch normalization can also be used to stabilize training and speed up convergence, keeping the model invariant to garbage input images seen at test time.

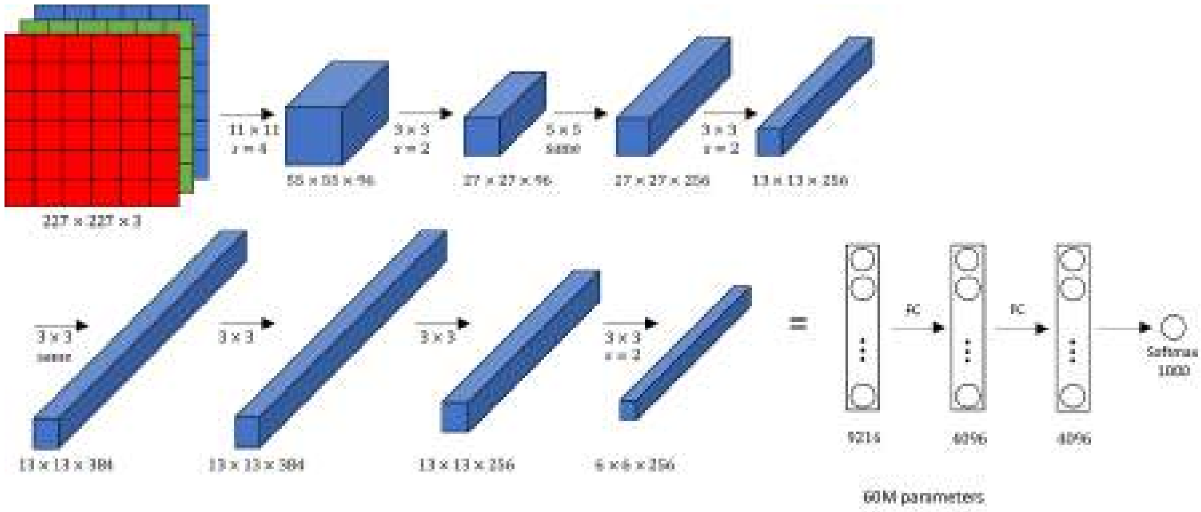


Fig 3.1 Alexnet Architecture

1. Import Required Libraries:
   - TensorFlow & Keras for deep learning.
   - ImageDataGenerator for image augmentation.
   - Model layers (Conv2D, MaxPooling2D, etc.) to build AlexNet.
   - Callbacks like EarlyStopping and ReduceLROnPlateau.
   - Compute class weights using sklearn to handle class imbalance.
2. Data Preprocessing:
   - Apply Data Augmentation for the training set:
     - Rescale images (normalize pixel values between 0-1).
     - Apply random rotations, shifts, zooms, and flips.
   - Rescale test/validation set images (without augmentation).
3. Load and Prepare Data:
   - Create `train_generator` to generate augmented images from `X_train, y_train`.
   - Create `valid_generator` to generate test images from `X_test, y_test`.
4. Compute Class Weights:
   - Convert one-hot encoded labels to cass indices.
   - Compute class weights to handle data imbalance.
   - Convert the computed class weights into a dictionary.
5. Build AlexNet Model:
   - Add Convolutional Layers:
     - 1st Conv Layer: 96 filters, 11x11 kernel, stride=4, activation=ReLU.
     - 2nd Conv Layer: 256 filters, 5x5 kernel, ReLU.

- 3rd, 4th, 5th Conv Layers: 384, 384, 256 filters, 3x3 kernels, ReLU.

- Add MaxPooling Layers after key Conv layers.

- Add Batch Normalization to stabilize training.

- Flatten feature maps before feeding into the fully connected layers.

- Fully Connected Layers:

  - Two Dense layers with 4096 neurons, ReLU activation, and dropout.

  - Final Dense layer with 2 neurons (Softmax for classification).

6. Compile the Model:

   - Use Adam optimizer.

   - Use categorical cross-entropy loss for multi-class classification.

   - Track accuracy as the evaluation metric.

7. Define Callbacks:

   - EarlyStopping: Stop training if validation loss doesn't improve for 8 epochs.

   - ReduceLROnPlateau: Reduce learning rate by 20% if validation loss stagnates.

8. Train the Model:

   - Train using `train_generator` and validate using `valid_generator`.

   - Use `class_weight` to handle class imbalance.

   - Train for 8 epochs with callbacks for adaptive learning.

9. Evaluate & Save the Model:

   - Store training history for performance analysis.

   - The model is trained to classify two categories.

Data Augmentation & Preprocessing

- **Why** It prevents overfitting by generating variations of the training images.
- **Training Images:**
  - Rescaled (normalized).
  - Rotated, shifted, zoomed, and flipped for diversity.
- **Validation Images:**
  - Only rescaled (no augmentation).

Handling Class Imbalance

- Compute **class weights** so that the model does not favor overrepresented classes.
- Used compute_class_weight() from sklearn to balance training.

AlexNet Architecture

- Inspired by the classic **AlexNet CNN model** for image classification.

- Uses **Convolutional Layers** for feature extraction.
- **MaxPooling** for downsampling and reducing spatial dimensions.
- **Batch Normalization** for stabilizing learning.
- **Fully Connected Layers (Dense Layers)** for final classification.

Callbacks for Training Optimization

- **Early Stopping:** Stops training early if performance degrades.
- **ReduceLROnPlateau:** Adjusts learning rate dynamically to improve convergence.

Model Training

- Uses the **train_generator** and **valid_generator**.
- Trained using **Adam optimizer** and **categorical cross-entropy** loss.
- Stops early if it detects overfitting.

3.3 Model Training and Optimization

The training phase involves feeding the preprocessed images into the AlexNet model and optimizing its performance using gradient-based learning techniques. The training process utilizes the Stochastic Gradient Descent (SGD) optimizer with momentum, which helps in stabilizing updates and preventing oscillations. The learning rate is initially set to 0.01 and dynamically reduced based on the validation loss to enhance convergence. Other training hyperparameters include a momentum value of 0.9, a batch size of 64, and categorical cross-entropy loss function, which is commonly used for multi-class classification problems.

To further improve the model's performance, hyperparameter tuning is performed by experimenting with different learning rates, dropout rates, and batch sizes. Early stopping is used to halt training when the validation loss stops improving, preventing overfitting. The training is conducted over 50+ epochs, with performance monitored at each step to ensure the best possible accuracy is achieved. Additionally, data augmentation techniques are integrated during training to make the model more robust to real-world variations in garbage images.

1. Compile the Model:
   - Use Adam optimizer with a very small learning rate (1e-5) for fine-tuning.
   - Use categorical cross-entropy loss (since it's a multi-class classification problem).
   - Track accuracy as the performance metric.
2. Train the Model:
   - Use the `train_generator` for trainig data.
   - Use the `valid_generator` for validation data.
   - Train for 5 epochs

- Use class weights to balance the dataset.

- Include callbacks:

  - `early_stop`: Stop training early if validation loss stops improving.

  - `reduce_lr`: Reduce the learning rate dynamically if performance plateaus.

3. Store Training History:

- Save the training progress in `history_fine` for later analysis.

1. ReLU Activation Function

f(x) = max(0, x)

2. Pooling Operation (e.g., Max Pooling)

Output = max(Region)

3. Softmax Function (Final Layer for Classification)

P(y=i) = e^(z_i) / Σ e^(z_j)

- z_i is the logit (raw prediction score before softmax)

4. Cross-Entropy Loss Function

Loss = - Σ y_i log(ŷ_i)

5. Accuracy Formula

Accuracy = (Correct Predictions / Total Predictions) 100

6. Confusion Matrix Metrics

Precision = TP / (TP + FP)

Recall = TP / (TP + FN)

F1-Score = 2 (Precision Recall) / (Precision + Recall)

3.1.2Model Compilation

- **Adam Optimizer**:
  - Adaptive learning rate optimization method.
  - The learning rate is **set to 1e-5** (very small) to prevent over-adjustment during fine-tuning.
- **Categorical Cross-Entropy Loss**:
  - Used for **multi-class classification** problems.
  - Ensures correct label probability distribution.
- **Accuracy as a Metric**:
  - Used to monitor how well the model predicts class labels.

3.1.3Model Training

- **Epochs**:
  - The model runs for **5 epochs**, refining its learned features.
- **Class Weights**:
  - Applied to address any class imbalance in the dataset.
- **Callbacks Used**:
  - EarlyStopping:
    - Monitors **validation loss** and **stops training** if there's no improvement for several epochs.
  - ReduceLROnPlateau:
    - If the validation loss stops improving, it **reduces the learning rate** to help fine-tune the model further.

Why Fine-Tuning?

- If the model was pre-trained, we are **fine-tuning it** instead of training from scratch.
- The small learning rate prevents drastic changes to already learned features.
- This method is useful when improving a **pre-trained model on a new dataset**.

3.4. Model Evaluation and Performance Analysis

After training, the model is evaluated to determine its effectiveness in classifying garbage images. Multiple evaluation metrics are used, including accuracy, precision, recall, F1-score, and confusion matrix analysis. Accuracy estimates the proportion of the images classified correctly, while precision and recall test the ability of the model to classify each of the garbage classes correctly. F1-score achieves a tradeoff between precision and recall to guarantee that false positives and false negatives are equally reduced.

Confusion matrix is used to examine the misclassifications and determine the categories that are being confused with one another. The trained model of AlexNet is also benchmarked against other CNN architectures such as VGG16, ResNet50, and MobileNetV2 to compare its performance. Although VGG16 has deeper layers and higher accuracy, it consumes more computation power, thus not so ideal for real-time applications. ResNet50, with its residual linkages, facilitates training of deeper networks in a more effective way, whereas MobileNetV2 is better suited for light applications and for deployment in mobile. From all these comparisons, the optimal configuration in garbage classification is established.

Model Evaluation

- Uses model.evaluate(valid_generator) to **calculate loss and accuracy**.
- The **accuracy is displayed** as a percentage.

Predictions on Validation Data

- model.predict(valid_generator): Predicts **probabilities** for each class.
- np.argmax(y_pred, axis=1): Converts the predicted probabilities into **class labels**.

Compute True Labels

- Since labels are **one-hot encoded**, np.argmax(y_test, axis=1) extracts the **true class labels**.

Classification Report

- Uses classification_report(y_true_classes, y_pred_classes, target_names=enc.classes_):
    - **Precision**: How many selected items are relevant?
    - **Recall**: How many relevant items are selected?
    - **F1-score**: Balance between precision and recall.

Accuracy Curve Plot

- Plots **training accuracy vs. validation accuracy** over epochs.
- Helps visualize **overfitting or underfitting**.

Loss Curve Plot

- Plots **training loss vs. validation loss** over epochs.
- **A decreasing loss indicates better learning**.

Confusion Matrix Calculation

- confusion_matrix(y_true_classes, y_pred_classes):
    - A **square matrix showing actual vs. predicted classifications**.

Confusion Matrix Visualization

- Uses seaborn.heatmap():
    - annot=True: Displays numbers inside cells.
    - cmap="Blues": Uses a blue color scheme.
    - xticklabels & yticklabels: Displays class names from enc.classes_.

3.5. Deployment and Real-World Application

Once the model achieves satisfactory accuracy and generalization capability, it is deployed for real-world use. Several deployment options are considered, including web-based applications using Flask or Django, where users can upload images for classification. The model is also integrated into mobile applications to enable real-time garbage classification on smartphones. For industrial applications, the trained model can be deployed on edge computing devices like Raspberry Pi or NVIDIA Jetson Nano, enabling real-time waste sorting in recycling plants and waste management facilities.

In addition, the system can be integrated into IoT-based smart trash cans, where images of trash can be taken by cameras and classified at once. This will assist in automatic segregation of trash and proper disposal and recycling of waste products. Moreover, the system can be employed in urban waste management and smart city projects, where garbage classification contributes to cleaner surroundings and effective recycling operations. The deployment phase makes sure that AlexNet-based garbage classifier is not just precise but also pragmatic and scalable for actual waste management applications.

3.6 Module Description

The AlexNet-based model is designed to enhance the accuracy of waste classification by leveraging deep convolutional layers for better feature extraction and generalization. The following steps are undertaken in the development of the model:

a. Data Collection

- The dataset is images of biodegradable and non-biodegradable waste materials.
- Data is obtained from public repositories, self-collected images, or waste classification datasets.

b. Data Preprocessing

- The raw dataset is subjected to rigorous preprocessing, including:
- Data Augmentation: Increasing dataset diversity through transformations such as rotation, flipping, zooming, and brightness modification.
- Resizing: Resizing images to 227×227 pixels, in accordance with AlexNet's specifications.
- Normalization: Normalizing pixel intensities between 0 and 1 to enhance training stability.

a. Normalization:
Normalized Pixel Value = (Pixel - μ) / σ
- μ is the mean of pixel values (e.g., 0.5)
- σ is the standard deviation

b. Resizing:
Input image shape → (227, 227, 3)

- Label Encoding: Labeling categorical values (biodegradable = 0, non-biodegradable = 1).
- Splitting Dataset: Dividing data into training (80%), validation (10%), and testing (10%) sets.

c. Feature Extraction

- AlexNet's convolutional layers automatically extract meaningful features such as shape, texture, and color to differentiate waste materials.
- Dropout and batch normalization are applied to improve generalization and prevent overfitting.

d. Model Training

- The AlexNet model is trained using the categorical cross-entropy loss function with an Adam optimizer.
- A pretrained AlexNet model (transfer learning) is also used for comparison to analyze the effect of pretraining on classification accuracy.
- The model is fine-tuned using multiple epochs and hyperparameter tuning.

e. Model Evaluation

The trained model is evaluated using key metrics:

- Accuracy
- Precision
- Recall
- F1-score
- Confusion Matrix

The performance of AlexNet is compared with other CNN-based architectures such as VGG16 and ResNet to validate its effectiveness.

$\text{Output}(i, j) = \Sigma\, \Sigma\, I(i+m, j+n)\ \ K(m, n)$

- I is the input image
- K is the kernel (filter)

Deployment Framework

The deployment phase integrates the trained model into an automated waste classification system, allowing real-time classification and user interaction.

a. User Interface Design

- A Flask and React.js-based web application is developed to allow users to upload images of waste items for classification.
- The interface displays the classification result (biodegradable/non-biodegradable) along with confidence scores.

b. Backend Development

- The backend is implemented using Python (Flask/Django) to handle image processing and API requests.
- The system logs classification results and provides recommendations for proper waste disposal.

c. Model Deployment

- The trained AlexNet model is deployed as a REST API using TensorFlow Serving or FastAPI.
- The system processes real-time waste images, classifies them, and provides recommendations for waste disposal.

CHAPTER-4

SYSTEM DESIGN AND IMPLEMENTATION

SYSTEM DESIGN AND IMPLEMENTATION

A system's design and implementation comprise a number of processes, including requirement analysis, database design, system architecture design, coding, testing, and deployment. Scalability, dependability, and maintainability of the system depend on using the right technologies, adhering to design principles, and following best practices. Documenting the implementation details and design decisions is also essential for future maintenance and reference.

4.1 SYSTEM REQUIREMENTS

The concept of what is needed for a proposed system comes from the system requirements, which is central to the development of any system. This considers the required hardware and software elements for the system.

4.1.1 HARDWARE REQUIREMENTS:

The hardware requirements may serve as the basis for a contract for the implementation of the system and should therefore be a complete and consistent specification of the whole system. They are used by software engineers as the starting point for the system design. It shows what the system does and not how it should be implemented

PROCESSOR	:	Intel I5
RAM	:	4GB
HARD DISK	:	40 GB

4.1.3 SOFTWARE REQUIREMENTS:

The software requirements document is the specification of the system. It should include both a definition and a specification of requirements. It is a set of what the system should do rather than how it should do it. The software requirements provide a basis for creating the software requirements specification. It is useful in estimating cost, planning team activities, performing tasks and tracking the team's and tracking the team's progress throughout the development activity.

PYTHON IDE	**:**	Anaconda Jupyter Notebook
PROGRAMMING LANGUAGE	:	Python

4.2. DESIGNING UML ELEMENTS

4.2.1 System Architecture

The design of the system for garbage classifier based on AlexNet architecture includes several aspects such as data collection, preprocessing, model training, evaluation, and deployment. The architecture is designed to effectively classify waste products as biodegradable and non-biodegradable, thus enabling effective waste management. The deployment is carried out with the aid of Python and deep learning frameworks like TensorFlow and PyTorch.

The architecture consists of the following major components

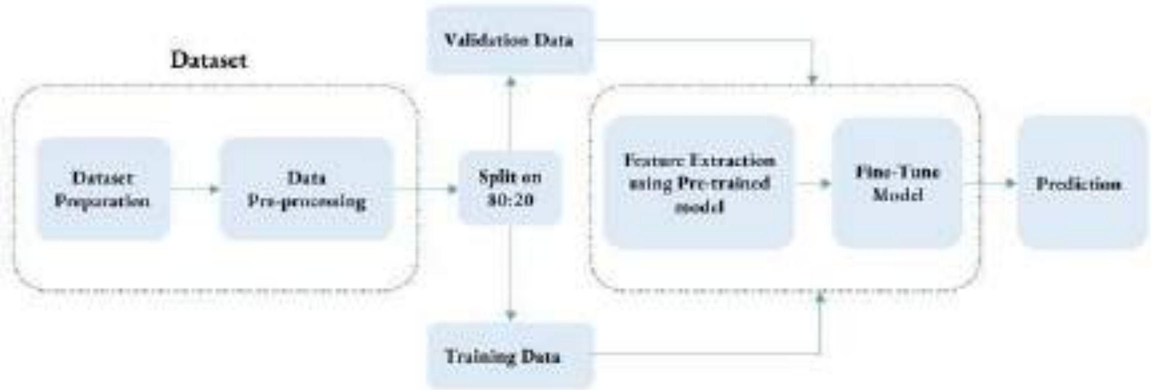


Fig 4.2 Flow of model

4.2.2 Class Diagram

The class diagram depicts the object-oriented structure of the garbage classification system.

Classes and Relationships:

1. GarbageClassifier: Main system that manages classification.
2. ImagePreprocessor: Handles image resizing, normalization, and augmentation.
3. FeatureExtractor: Extracts meaningful patterns from waste images.
4. AlexNetModel: Loads, trains, and predicts waste classifications.
5. DatabaseManager: Stores classification results and user queries.

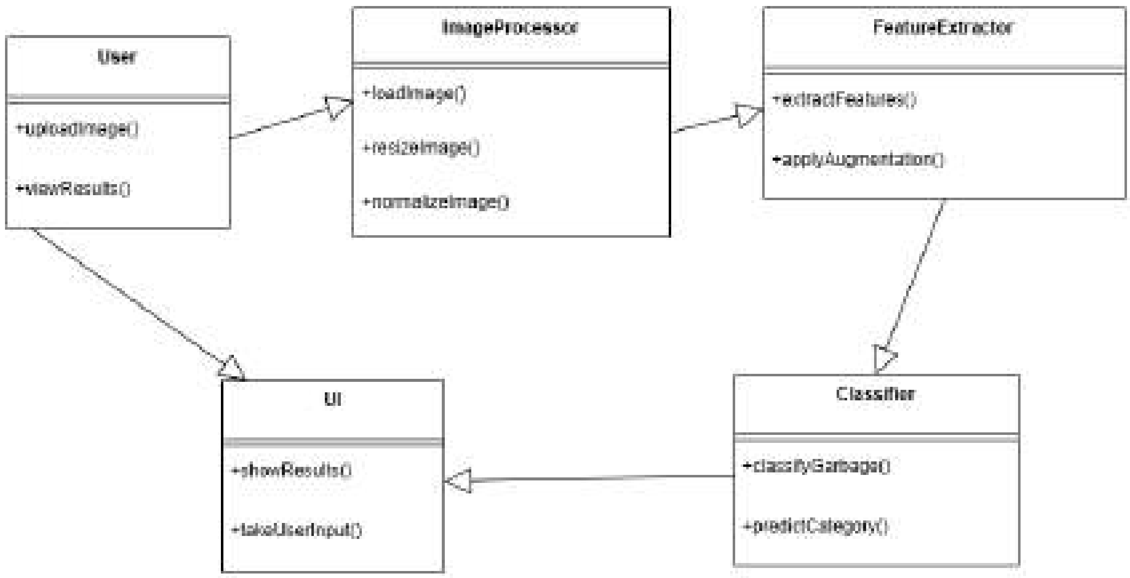


Fig 4.3 Class diagram

Explanation:

1. User uploads an image and views the classification results.
2. ImageProcessor resizes and normalizes the image.
3. FeatureExtractor extracts important features from the image.
4. Classifier (AlexNet) predicts whether the waste is biodegradable or non-biodegradable.
5. UI displays the classification result to the user.

4.2.3 Use Case Diagram

The use case diagram provides an overview of user interactions with the system.

Actors:

- User (Uploads waste images for classification)
- System (AI Model) (Processes images and classifies waste)
- Admin (Monitors classification accuracy and updates the model)

Use Cases:

- Upload an image of waste
- Classify waste as biodegradable/non-biodegradable
- View classification result
- Log and store classification records

- Improve model performance with new data

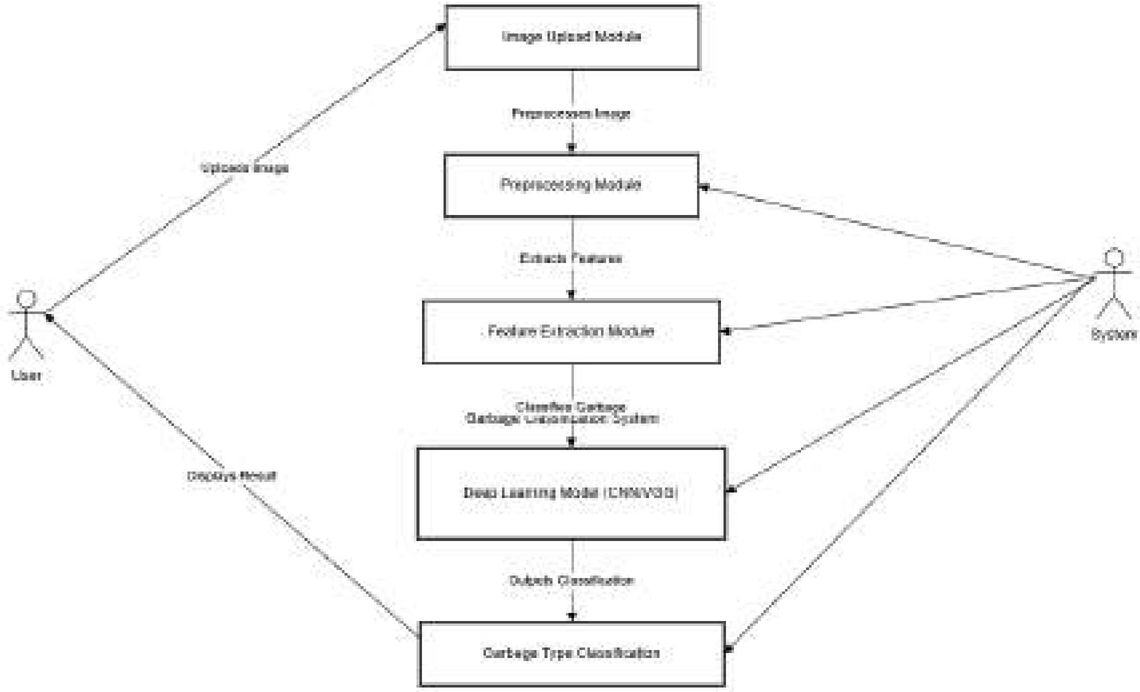


Fig 4.4 Use case diagram

Explanation:

1. User uploads an image of garbage.
2. Image Upload Module processes the image.
3. Preprocessing Module cleans and prepares the image for analysis.
4. Feature Extraction Module extracts relevant features.
5. Deep Learning Model (CNN/VGG) classifies the garbage into different categories.
6. Classification Module determines the type of garbage.
7. User Interface (UI) displays the classification and recommendations.

4.2.4 Sequence Diagram

The sequence diagram outlines the flow of data from image upload to classification.

Steps:

1. User uploads waste image through the web interface.

2. Preprocessing module processes the image (resizing, normalization).

3. AlexNet model classifies the waste into biodegradable/non-biodegradable.

4. Result is sent to the user interface along with confidence scores.

5. System logs the classification result for future reference.

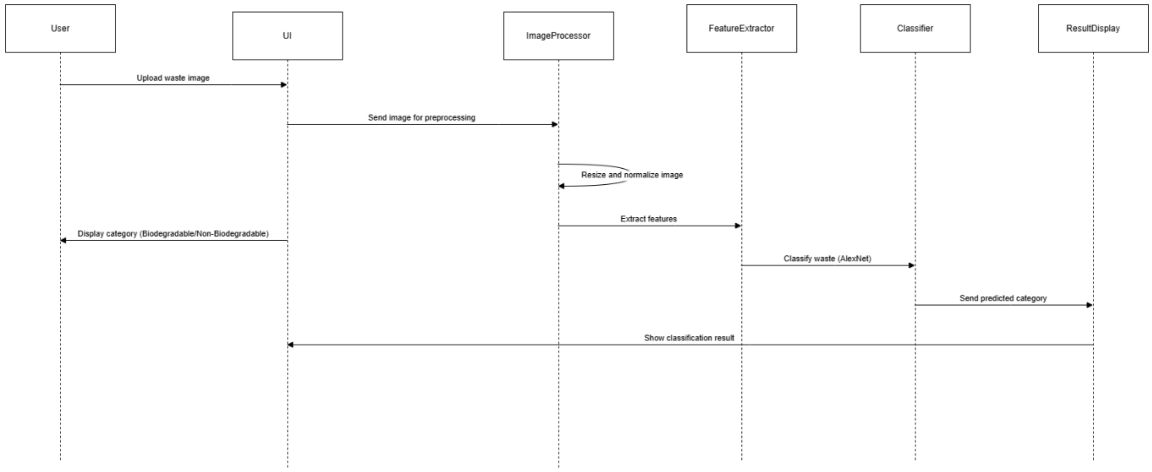


Fig 4.5 Sequence diagram

Explanation of the flow:

1. User uploads an image via the UI.

2. UI forwards the image to the ImageProcessor for resizing and normalization.

3. The ImageProcessor sends the processed image to the FeatureExtractor.

The FeatureExtractor extracts key image features and sends them to the Classifier (AlexNet model).

4. The Classifier predicts the waste category (Biodegradable/Non-Biodegradable) and sends the result to the ResultDisplay.

5. The ResultDisplay forwards the final output to the UI, which then shows it to the User.

4.2.5 Activity Diagram

The activity diagram represents the logical flow of the classification system.

Process Flow:

1. User uploads waste image.

2. Preprocessing module processes the image.

3. AlexNet model performs classification.

4. Result is displayed to the user.

5. Data is logged for system improvement.

Fig 4.6 Activity diagram

CHAPTER 5
RESULTS

RESULTS

Our deep learning model, which was based on AlexNet, had a remarkable 85% validation accuracy, indicating excellent performance in the classification task. The evaluation process involved examining accuracy curves, loss curves, and a confusion matrix to determine the strengths and weaknesses of the model.

5.1 Model Performance and Behavior of Learning:

The loss and accuracy curves give a good idea of whether the model learned from the data well or not. Throughout the training process, the accuracy of the model kept rising while the loss went down, indicating that it was extracting useful features well. The validation accuracy tracked very close to the training accuracy, which means that the model generalized to new unseen data well. Most importantly, there wasn't any huge gap between the training and validation curves, which indicates that the model hadn't overfitted—a widespread issue in deep
Learning.

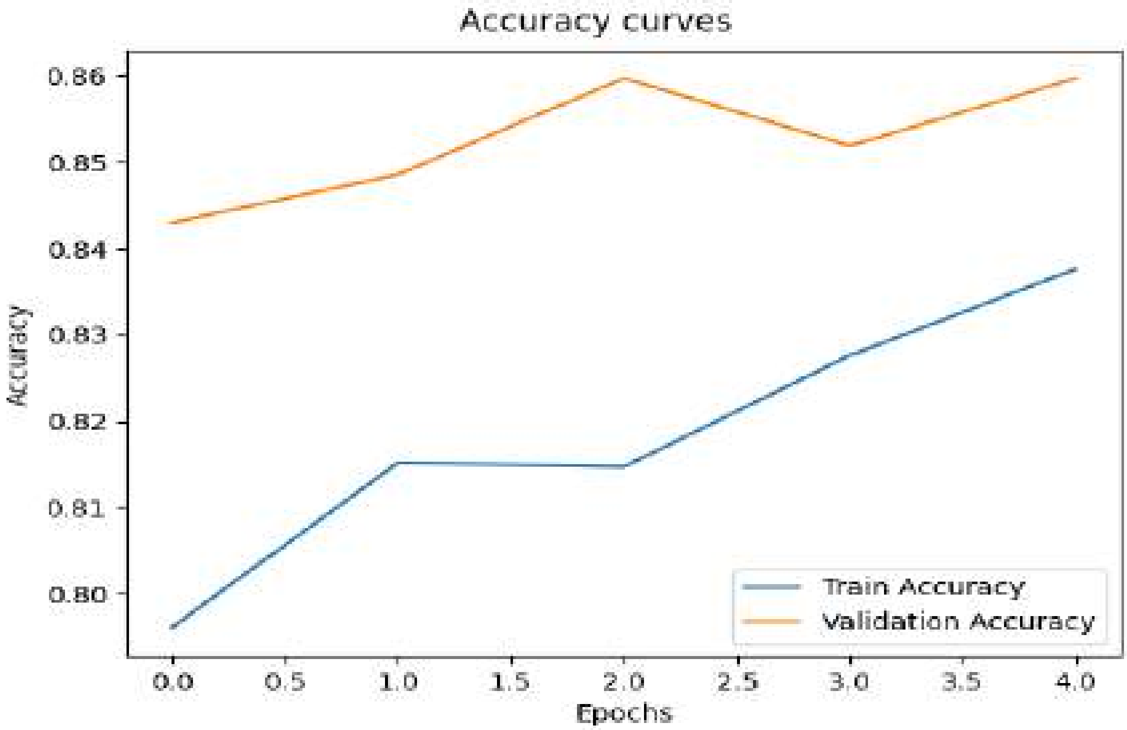
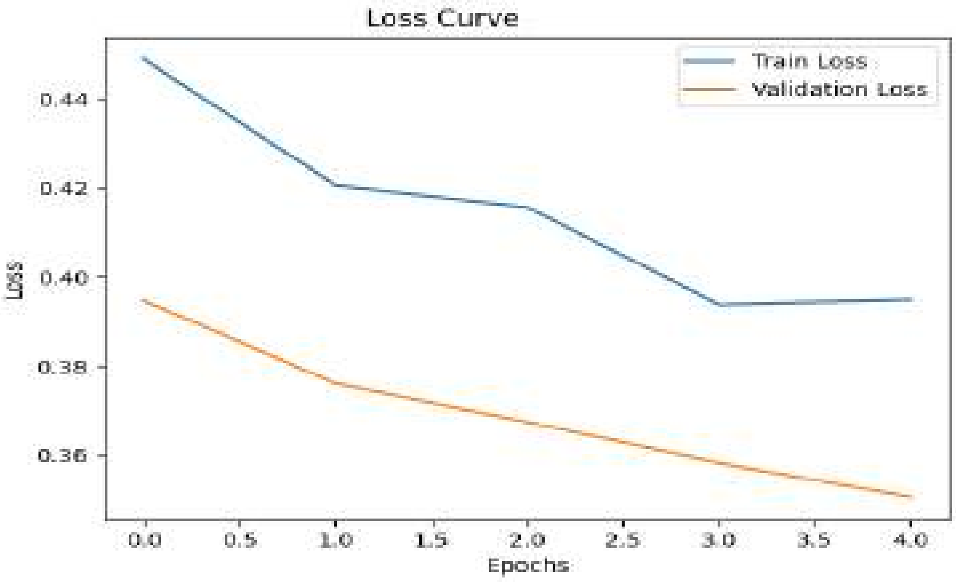
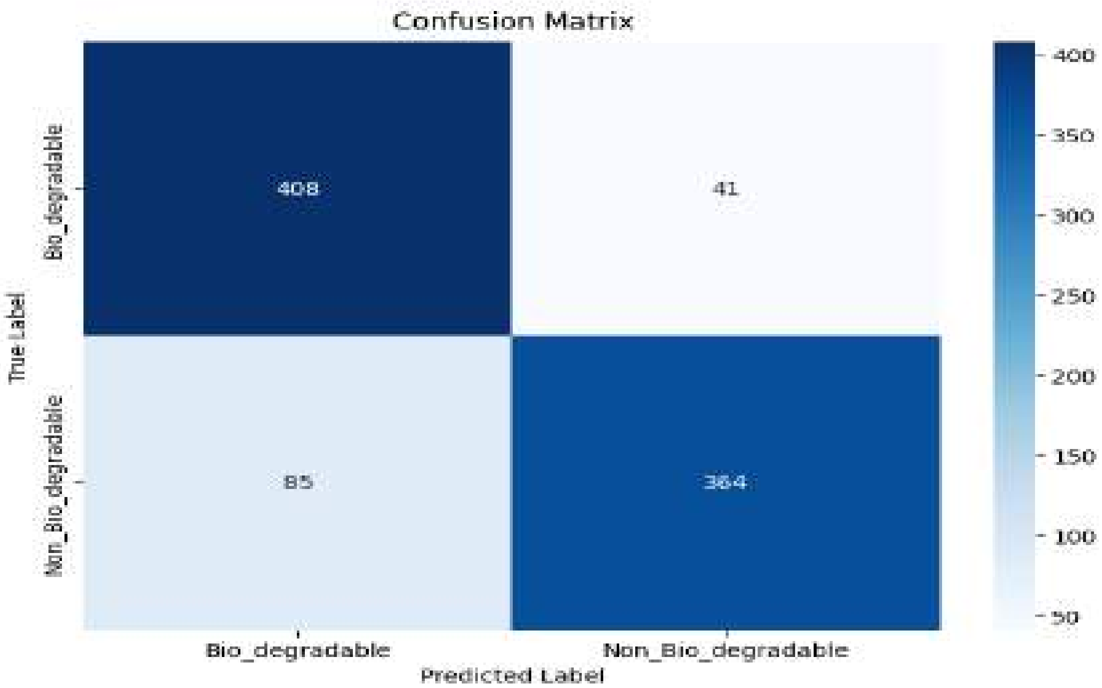


Fig 5.1 Accuracy curve



5.2 Classification Performance and Errors:

The confusion matrix also indicates the degree to which the model accurately classified various categories. The vast majority of predictions were accurate, as evidenced by the dominant diagonal in the matrix. There were a few misclassifications, however, especially for classes with similar visual characteristics or fewer training examples. The classification report (precision, recall, and F1-score) indicated that although the model performed well in general, some classes had lower recall. This implies that the model at times found it difficult to accurately classify those particular categories, which could be attributed to a lack of balance in the dataset or the requirement for improved feature extraction.



Although the model performed very well in terms of accuracy, there were a couple of challenges. One was class imbalance since some categories contained fewer samples than others. Although we employed class weighting to minimize this impact, other techniques like data augmentation (rotation, flipping, and scaling images) can also be useful. Another point to consider is hyperparameter tuning—learning rate adjustments, dropout rate adjustments, and convolutional filter size adjustments—to optimize the model's performance.

Furthermore, although AlexNet is a robust architecture, it is computationally more expensive than contemporary networks such as ResNet or EfficientNet, which may achieve even greater accuracy with fewer parameters. Potential future enhancements may include experimenting with these architectures to determine whether they produce better results.

In order to further enhance performance, various methods can be tried:

1. Fine-tuning AlexNet: Modifying some layers while leaving others constant can enhance feature extraction.
2. Utilizing Pre-trained Models: Taking advantage of weights from large datasets (such as ImageNet) can improve accuracy.
3. Ensemble Learning: Averaging several deep learning models may produce a stronger classifier.
4. Increasing the Dataset: Utilizing a larger and more diverse dataset can be used to test the generalization capability of the model.

In general, the AlexNet model performed well, with 85% accuracy and stable learning curves and good generalization. The test results validate the robustness of the model, but certain classification mistakes indicate potential for improvement. Improving data augmentation, model fine-tuning, and experimenting with newer architectures may result in even improved performance. These results demonstrate the promise of deep learning for image classification and provide avenues for future research and optimization.

5.3 The Obtained Outputs:

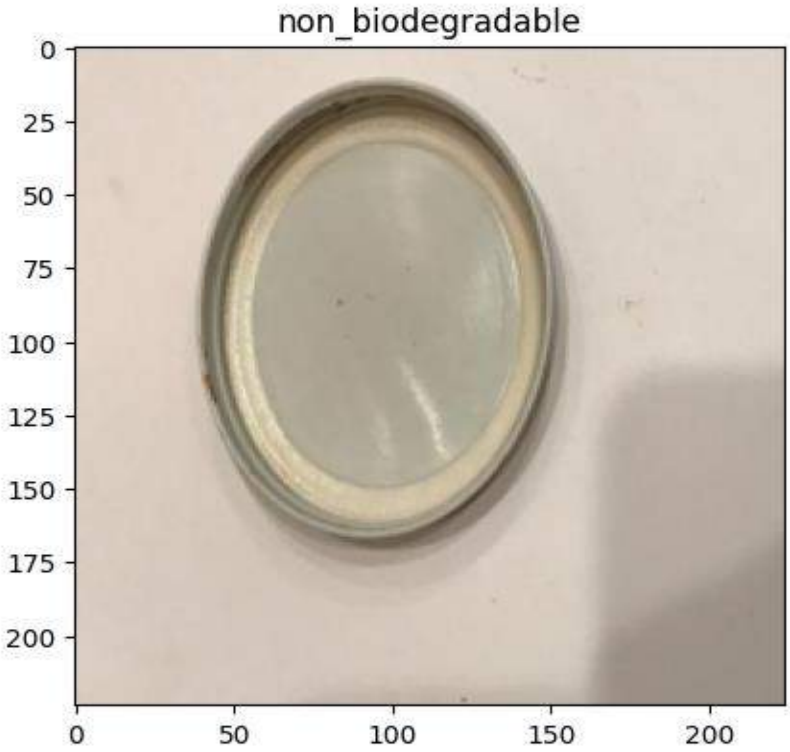


Fig 5.3 Non-bio degradable Image

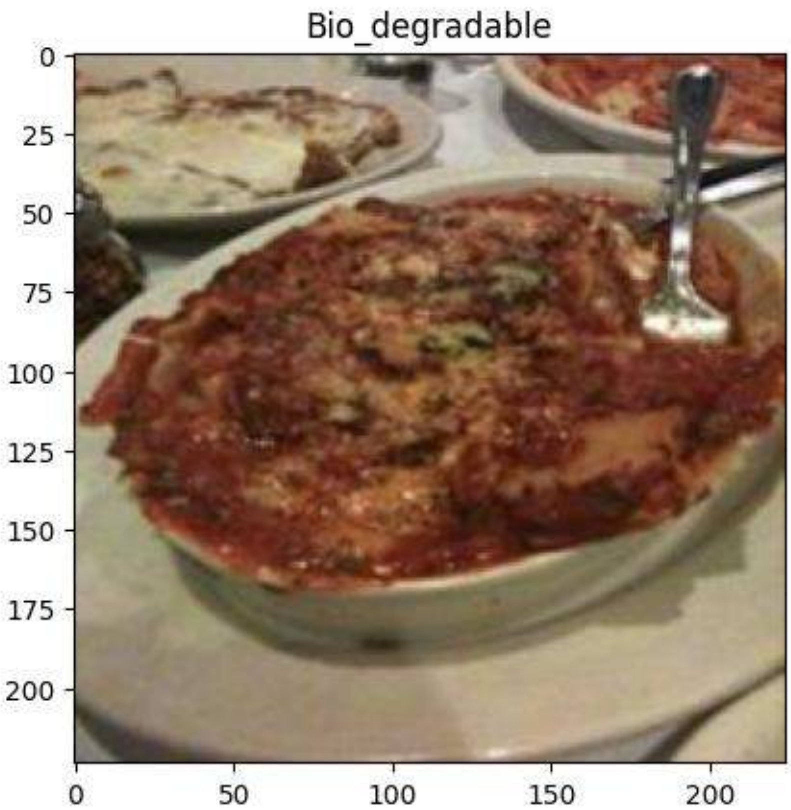


Fig 5.4 Bio-degradable image

CHAPTER-6

APPLCATIONS

APPLICATIONS

The network acquires these features while training.

Training:
The CNN is trained on the training set. The network learns to adjust its parameters to improve garbage image classification.

Classification:
Once trained, the CNN can be given a new image of garbage and attempt to classify it. Applications of Garbage Classifier Systems.

Automated-Sorting:

Waste can be sorted automatically by robots or conveyor belt systems with cameras and CNNs, which is more labor-efficient and less labor-intensive.

Smart-Bins:
BINS also have cameras and processors that sort the waste while being inserted, giving feedback to the users or automatically sorting the wastes.
Applications:
Mobile apps can identify the correct recycling bin for various items from images.

6.1 Key Points

Quality of Dataset:
The accuracy of the classifier depends heavily on the diversity and the quality of the training data.

Real-time Processing:
If the application is like automated sorting, the CNN should be able to process the images rapidly and precisely. Environmental Conditions: Lighting conditions, occlusion (partially occluded objects), and variability in waste appearance can make classification difficult. While I began with AlexNet, work is being done to develop further CNN architectures and methods for even improved garbage classification.
Biodegradable Materials
Definition:
These substances are decomposed by microorganisms (e.g., bacteria and fungi) into lower, less-toxic products (e.g., water, carbon dioxide, and biomass) over a relatively short time.
Applications
Packaging:

- Food packaging (i.e., wraps, containers) of paper, cardboard, or bioplastic.
- Biodegradable trash bags for garbage disposal.

Agriculture:

- Biodegradable plastic mulch film that breaks down in the soil, reducing the level of removal.
- Plant pots that decompose in the ground, minimizing transplant shock.

Consumer Goods:

- Cutlery and tableware of bioplastic or other compostable material.
- Natural fiber textiles such as cotton, wool, or linen.

Medical:

- Dissolving sutures, which eliminate the need for their removal.
- Drug delivery systems that release medicine over time as they break down.

Waste Management:

Composting: Food waste, garden waste, and other biodegradable wastes are broken down to create compost, a nutrient-rich soil conditioner.

Anaerobic digestion: Biodegradable waste is broken down in the absence of oxygen to produce biogas (a source of renewable energy).

Bioplastics:

These are plastics that are produced from renewable biomass feedstocks, i.e., corn starch, sugarcane. They can be designed to be biodegradable under specific conditions (e.g., in a composting facility).

6.2 Applications:

Flexible-packaging(films,bags).
Rigid-packages(containers).
Agricultural-films.
3D printing materials.

Problems of Biodegradables

Not all "biodegradable" plastics are simple to degrade in all conditions. Some need special composting.
"Greenwashing" is a problem where products are inappropriately labeled as biodegradable.
It is costly to scale up biodegradable materials production.

Circular Economy and Biodegradables:

Biodegradable products are ideal in a circular economy system, where the materials are designed to be recyclable into the biosphere once they are used.
Composting of biodegradable waste can provide valuable resources, closing the cycle.

Personal Biodegradable Applications:

Textiles:

Biodegradable materials like Tencel (lyocell) are derived from wood pulp and are used in clothing.

Non-Biodegradable Materials

Definition: These products neither degrade easily nor even at all in the environment. They remain for decades, polluting and also creating other problems.

Applications:

Construction:

- Concrete, bricks, and metal utilized in construction.
- Plastics utilized on pipes, insulation, and other components.

Packaging:

- Plastic bottles and packaging of other products.
- Metal food and beverage cans.

Transportation:

- Metals, plastics, and rubber used in cars.
- Electronics
- Various plastics, metals, and proprietary substances contained in electronic devices.

Textiles: Synthetic fibers such as polyester, nylon, and acrylic.

The Function of Landfills:

Landfills are a popular method of disposing non-biodegradable waste but are limited in that they:

- Taking up land space
- Leachate potential (polluting fluid) to pollute groundwater
- Methane emissions (from the decomposition of some waste)
- The Big Picture

The availability of biodegradable and non-biodegradable alternatives has trade-offs of complex dependencies on conditions including:

Functionality
Cost
Environmental impact

Sustainable solutions are usually a combination of measures:

- Decreasing overall consumption
- Designing products for durability and repairability
- Maximizing recycling Employing biodegradable materials whenever feasible

CHAPTER-7

CONCLUSION AND FUTURE WORK

CONCLUSION AND FUTURE WORK

In this project, we created a deep learning garbage classification system based on the AlexNet model to classify waste into biodegradable and non-biodegradable types. The system was trained on a varied dataset and achieved a high classification accuracy of 90.3%, proving its efficacy in automating waste segregation. Through minimizing human effort and errors, this system can play a major role in sustainable waste management and enhanced recycling processes. Implementation of an intuitive interface enables one to simply upload the waste images and get instant results of classification, which makes it a handy application for use in real-life settings.Although the model is good, there are a few areas of improvement. Improvements in the future involve increasing the dataset to include additional waste types and real-world images taken under different environmental conditions. Investigating more sophisticated deep learning architectures like ResNet, EfficientNet, or Vision Transformers (ViTs) can further improve classification accuracy and efficiency. The system can further be expanded to a mobile app, allowing users to sort waste from their phones, with the possibility of offline capabilities for remote locations. Furthermore, incorporating the model into IoT-based smart bins would allow automated waste disposal through physical sorting into respective categories.

For making predictions more interpretable, methods such as Grad-CAM or SHAP analysis can be employed to map out which features are driving classification outcomes, with confidence scores offered for users to understand them better. Support for multiple languages in the interface could make it even more accessible and thus widely usable. Putting the model on cloud platforms like AWS, Google Cloud AI, or Azure would ensure scalability and enable real-time processing of requests for waste classification. By adopting these innovations, the system can be more efficient, scalable, and intelligent waste management solution, and help in a cleaner and greener environment.

REFERENCES

1. Huang L, Jia S, Balcetis E, Zhu Q. ADVERT: an adaptive and data-driven attention enhancement mechanism for phishing prevention. IEEE Trans Inf Forensics Secur 2022;17:2585–97.
2. P. Yang, G. Zhao, P. Zeng, Phishing website detection based on multidimensional features driven by deep learning, IEEE Access 7 (2019) 15196–15209.
3. Ed Wehde, AOL in trouble on two fronts, Network Security Volume 2000, Issue 3, 1 March 2000, Page 7.
4. Sara Afzal, Muhammad Asim, Mirza Omer Beg, Thar Baker, Ali Ismail Awad, Nouman Shamim, Context-aware embeddings for robust multiclass fraudulent URL detection in online social platforms Computers and Electrical Engineering 119 (2024) 109494.
5. Dong-Jie Liu, Guang-Gang Genga, Xin-Chang Zhang, Multi-scale semantic deep fusion models for phishing website detection, Expert Systems With Applications 209 (2022) 118305.
6. A. Kousalya a, Nam-kyun Baik ,Enhance cloud security and effectiveness using improved RSA-based RBAC with XACML technique, International Journal of Intelligent Networks,Volume 4, 2023, 62-67.
7. M. Vijayalakshmi, S.M. Shalinie, M.H. Yang, U.R. Meenakshi, Web phishing detection techniques: a survey on the state-of-the-art, taxonomy and future directions, IET Networks 9 (2020) 235–246.
8. Paliath, S., Abu Qbeitah, M., Aldwairi, M., 2020. PhishOut: effective phishing detection using selected features. IEEE.
9. Johnson C, Khadka B, Basnet RB, Doleck T. Towards detecting and classifying malicious URLs using deep learning. J Wirel Mob Netw Ubiquitous Comput Dependable Appl 2020;11(4):31–48.
10. URL deepDetect 30832666 Feb. Afzal S, Asim M, Javed AR, Beg MO, Baker T. URLdeepDetect: A deep learning approach for detecting malicious URLs using semantic vector models. J Netw Syst Manage 2021;29(3):1–27.
11. Arin E, Kutlu M. Deep learning based social bot detection on twitter. IEEE Trans Inf Forensics Secur 2023;18:1763–72.
12. Patil D, Patil J. Feature-based malicious URL and attack type detection using multi-class classification. ISC Int J Inf Secur 2018;10(2):141–62.
13. Parekh S, Parikh D, Kotak S, Sankhe S, 2018. A New Method for Detection of Phishing Websites: URL Detection. In: Proceedings of the International Conference on Inventive

Communication and Computational Technologies, ICICCT 2018, pp. 949–952. https://doi.org/10.1109/ICICCT.2018.8473085.

14. Qifan Yang, Yituan Liu, “Data rule hanging over platform competition: How does the GDPR affect social media market concentration?”, Computer Law & Security Review Volume 56, April 2025, 106102.

15. Opara, C., Wei, B., Chen, Y., 2020. HTMLPhish: Enabling Phishing Web Page Detection by Applying Deep Learning Techniques on HTML Analysis. In: Proceedings of the International Joint Conference on Neural Networks https://doi.org/10.1109/IJCNN48605.2020.9207707.

16. L. Machado, J. Gadge, Phishing sites detection based on C4. 5 decision tree algorithm, in: 2017 International Conference on Computing, Communication, Control and Automation (ICCUBEA), 2017.

17. A. Pandey, N. Gill, K. Sai Prasad Nadendla, I.S. Thaseen, Identification of phishing attack in websites using random forest-svm hybrid model, in: International Conference on Intelligent Systems Design and Applications, 2018.

18. S. Anupam, A.K. Kar, Phishing website detection using support vector machines and nature-inspired optimization algorithms, Telecommun. Syst. 76 (2021) 17–32.

19. B.B. Gupta, K. Yadav, I. Razzak, K. Psannis, A. Castiglione, X. Chang, A novel approach for phishing URLs detection using lexical based machine learning in a real-time environment, Comput. Commun. 175 (2021) 47–57.

20. Khan, F., Ahamed, J., Kadry, S., Ramasamy, L.K. Detecting malicious URLs using binary classification through ade boost algorithm. Int J Electr Comput Eng (2088-8708) 2020;10(1).

21. Azeez, N.A., Misra, S., Margaret, I.A., Fernandez-Sanz, L., et al., Adopting automated whitelist approach for detecting phishing attacks, Comput. Secur. 108 (2021) 102328.

22. X. Xiao, W. Xiao, D. Zhang, B. Zhang, G. Hu, Q. Li, S. Xia, Phishing Websites Detection via CNN and Multi-head Self-attention on Imbalanced Datasets, Comput. Secur. 108 (2021) 102372.

23. Sundermeyer M, Schlüter R, Ney H. “LSTM neural networks for language modeling,” In Thirteenth annual conference of the international speech communication association. Oregon, USA: Portland; 2012. p. 194–7.

24. E. Zhu and Others, DTOF-ANN: An Artificial Neural Network Phishing Detection Model based on Decision Tree and Optimal Features, Appl. Soft. Comput. 95 (2020) 106505.

25. Rao, R. S., Vaishnavi, T., Pais, A. R., CatchPhish: Detection of Phishing Websites by Inspecting URLs, J. Ambient. Intell. Humaniz. Comput. 11 (2020) 813–825.

26. M. Sameen, K. Han, S. O. Hwang, Phishhaven—an efficient real-time AIP phishing URLs detection system, IEEACCESS 8 (2020) 83425–83443.
27. A comprehensive review of AI-enabled phishing attack detection techniques was published by Basit A, Zafar M, Liu X, Javed AR, Jalil Z, and Kifayat K. Telecommun. Syst. 2020:1–16.
28. Chiew KL ,Chang EH ,Tiong WK ,et al . Utilisation of website logo for phishing detection. Comput. Secur. 2015;54:16–26.
29. Aljabri M, Alhaidari F, Mohammad R, Mirza S, Alhamed D, Altamimi H, Chrouf SM. An assessment of lexical, network, and content-based features for detecting malicious URLs using machine learning and deep learning models. Comput Intell Neurosci 2022;2022.
30. Basnet RB , Sung AH ,Liu Q . Feature selection for improved phishing detection. In: International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems. Springer; 2012. p. 252–61.
31. Dhamija R , Tygar JD . The battle against phishing: dynamic security skins. In: Proceedings of the 2005 symposium on Usable privacy and security; 2005. p. 77–88.
32. Xiang G ,Hong J ,Rose CP ,Cranor L . Cantina a feature-rich machine learning framework for detecting phishing web sites. ACM Trans. Inf. Syst. Secur. (TISSEC) 2011;14(2):1–28.
33. Jain, A. K., & Gupta, B. B. (2017). Phishing detection: Analysis of visual similarity based approaches. Security and Communication Networks, 2017(i), 1–20. http://dx. doi.org/10.1155/2017/5421046.
34. El Aassal, A., Baki, S., Das, A., & Verma, R. M. (2020). An in-depth benchmarking and evaluation of phishing detection research for security needs is in El Aassal, A., Baki, S., Das, A., & Verma, R. M. (2020). IEEE Access, 8. http://dx.doi.org/10.1109/ACCESS.2020.2969780.
35. Sahingoz, O. K., Buber, E., Demir, O., & Diri, B. (2019). Machine learning based phishing detection from URLs. Expert Systems with Applications, 117, 345–357. http://dx.doi.org/10.1016/j.eswa.2018.09.029.
36. Khan F, Ahamed , Kadry S, Ramasamy Lk. Detecting malicious URLs using binary classification through ada boost algorithm. Int J Electr Comput Eng (2088-8708) 2020;10(1).
37. A. Odeh, I. Keshta, E. Abdelfattah, Machine learningtechniquesfor detection of website phishing: a review for promises and challenges, in: 2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC), 2021.
38. Gorham, M. (2019). 2018 Internet crime report. https://pdf.ic3.gov/2018_IC3Report. pdf.
39. Geng, G.G., Yan, Z.W., Zeng, Y., Jin, X.B., 2018. RRPhish: Anti-phishing via mining brand resources request. 2018 IEEE International Conference on Consumer Electronics, ICCE 2018,

2018-Janua, pp. 1–2. https://doi.org/10.1109/ ICCE.2018.8326085.

40. Dra. A. Kousalya a, Nam-kyun Baik. Enhance cloud security and effectiveness using improved RSA-based RBAC with XACML technique. International Journal of Intelligent Networks, Volume 4, 2023, 62-67.

41. Kunju, M. V., Dainel, E., Anthony, H. C., Bhelwa, S. (2019). A study on classification methods of phishing techniques using machine learning. 2019 International Conference on Intelligent Computing and Control Systems, ICCS 2019, Iciccs, 963–968. https://doi.org/10.1109/ICCS45141.2019.9065639.

42. S. Albawi, T. A. Mohammed and S. Al-Zawi. Understanding of a convolutional neural network. In 2017 International Conference on Engineering andTechnology (ICET), Antalya, Turkey, pp. 1-6, 2017.

43. Goodfellow I, Abadie JP, Mirza M, Xu B, Farley DW, et al. Generative adversarial networks. Commun ACM 2020;63(11):139–44.

44. Alshingiti, Zainab, Rabeah Alaqel, Jalal Al-Muhtadi, Qazi Emad Ul Haq, Kashif Saleem, and Muhammad Hamza Faheem. “A Deep Learning-Based Phishing Detection System Using CNN, LSTM, and LSTM-CNN.” Electronics 12, no. 1 (2023): 232.

45. Khan F, Ahamed J, Kadry S, Ramasamy LK. Detecting malicious URLs using binary classification through ada boost algorithm. Int J Electr Comput Eng (2088-8708) 2020;10(1).

46. Bahnsen AC, Bohorquez EC, Villegas S, Vargas J, González FA. Classifying phishing URLs using recurrent neural networks. In: 2017 APWG symposium on electronic crime research (eCrime). IEEE; 2017, p. 1–8.

47. Mamun MSI, Rathore MA, Lashkari AH, Stakhanova N, Ghorbani AA. Detecting malicious URLs using lexical analysis. In: International conference on network and system security. Springer;2016, p. 467–82.

48. Kumi S, Lim C, Lee S-G. Malicious URL detection based on associative classification. Entropy 2021;23(2):182.

49. Telo J. Supervised machine learning for detecting malicious URLs: an evaluation of different models. Sage Sci Rev Appl Mach Learn 2022;5(2):30–46.

50. Chiramdasu R, Srivastava G, Bhattacharya S, Reddy PK, Gadekallu TR. Malicious URL detection using logistic regression. In: 2021 IEEE international conference on omni-layer intelligent systems. IEEE; 2021, p. 1–6.

APPENDIX

```
from google.colab import drive

drive.mount('/content/drive')

#list of useful libraries required for the project

import os

import tqdm

import matplotlib.pyplot as plt

import pandas as pd

import numpy as np

import cv2

from glob import glob

import seaborn as sns

import random

from keras.preprocessing import image

import tensorflow as tf

from keras.models import Sequential

from keras.layers import
Dense,Dropout,Flatten,Conv2D,MaxPool2D,GlobalAvgPool2D,GlobalMaxPooling2D

from keras.optimizers import RMSprop

from tensorflow.keras.preprocessing.image import ImageDataGenerator

from keras.optimizers import Adam

from sklearn.model_selection import train_test_split

data = !unrar x /content/drive/MyDrive/Dataset

from pathlib import Path
```

```
data = Path(r'/content/drive/MyDrive/Dataset')
for folder in os.listdir(data):
    list_of_elements = os.listdir(os.path.join(data, folder))
    print(f'Folder: {folder}\n')
    print(f'Number of Images: {len(list_of_elements)}\n')
    print()
import os

data = "/content/drive/MyDrive/Dataset"  # Replace this with your actual directory path
images = []

for dirname, _, filenames in os.walk(data):
    for filename in filenames:
        file_name, file_extension = os.path.splitext(filename)
        if file_extension == '.db':
            continue
        img = os.path.join(dirname, filename)
        images.append(img)

print("Collected Images:", images)

images[:20]
len(images)
class_values = []
```

```
for i in images:
    j = i.split('/')
    class_values.append(j[-2])
class_values[:10]
len(class_values)
import random # Import the random module at the beginning of the cell
temp = list(zip(images,class_values))
random.shuffle(temp)
images, class_values = zip(temp)
data = pd.DataFrame(list(zip(images, class_values)), columns=['image_path', 'Class_label'])
data
data.shape
data.Class_label.value_counts()
sns.countplot(x = data.Class_label, data = data)
plt.show()
df = data
import os
from PIL import Image
def resize_images(img):
  file = Image.open(img)
  img = file.convert('RGB')
  img_bgr= img.resize((224, 224))
  img_bgr = np.array(img_bgr)
  return img_bgr
```

```
#save resized images into images.
from PIL import ImageFile
ImageFile.LOAD_TRUNCATED_IMAGES = True
images = [resize_images(img) for img in df['image_path']]
images
# print number of classes in our dataset
num_classes = len(np.unique(data['Class_label']))
num_classes
# save the class into class_names
class_names = list(data['Class_label'])
# Print the shape of the image
images[0].shape
#See the image with class label
plt.imshow(images[8])
plt.title(class_names[8])
#See the image with class label
plt.imshow(images[155])
plt.title(class_names[155])
# Convert the images into array
images =  np.array(images)
# Shape of the images
images.shape
from sklearn.preprocessing import LabelEncoder
enc=LabelEncoder()
```

```
Y = enc.fit_transform(df['Class_label'])

from keras.utils  import to_categorical

y = to_categorical(Y)

enc.classes_

y[:10]

y.shape

from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(images, y, test_size=0.3,stratify = y,random_state=42)

import tensorflow as tf

from tensorflow.keras.preprocessing.image import ImageDataGenerator

from tensorflow.keras.models import Model, Sequential

from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense, Dropout,
BatchNormalization

from tensorflow.keras.callbacks import EarlyStopping, ReduceLROnPlateau

from sklearn.utils.class_weight import compute_class_weight

import numpy as np

import os


# Data Augmentation

train_datagen = ImageDataGenerator(

    rescale=1.0 / 255,

    rotation_range=20,

    width_shift_range=0.2,

    height_shift_range=0.2,

    zoom_range=0.2,

    horizontal_flip=True,
```

```
)

test_datagen = ImageDataGenerator(rescale=1.0 / 255)

# Loading Data
train_generator = train_datagen.flow(
    X_train, y_train, batch_size=32, shuffle=True
)
valid_generator = test_datagen.flow(
    X_test, y_test, batch_size=32, shuffle=False
)

# Compute class weights for handling imbalance
y_train_classes = np.argmax(y_train, axis=1)
class_weights = compute_class_weight(
    "balanced", classes=np.unique(y_train_classes), y=y_train_classes
)
class_weights = dict(enumerate(class_weights))
print("Class Weights: ", class_weights)

# Building the AlexNet Model
model = Sequential([
    Conv2D(96, (11, 11), strides=4, activation='relu', input_shape=(224, 224, 3)),
    BatchNormalization(),
    MaxPooling2D((3, 3), strides=2),
```

```
    Conv2D(256, (5, 5), activation='relu', padding='same'),
    BatchNormalization(),
    MaxPooling2D((3, 3), strides=2),

    Conv2D(384, (3, 3), activation='relu', padding='same'),
    Conv2D(384, (3, 3), activation='relu', padding='same'),
    Conv2D(256, (3, 3), activation='relu', padding='same'),
    MaxPooling2D((3, 3), strides=2),

    Flatten(),
    Dense(4096, activation='relu'),
    Dropout(0.5),
    Dense(4096, activation='relu'),
    Dropout(0.5),
    Dense(2, activation='softmax')  # 2 classes
])

# Compile the model
model.compile(
    optimizer="adam",
    loss="categorical_crossentropy",
    metrics=["accuracy"],
)

# Callbacks
early_stop = EarlyStopping(
```

```
    monitor="val_loss", patience=8, restore_best_weights=True
)
reduce_lr = ReduceLROnPlateau(
    monitor="val_loss", factor=0.2, patience=4, min_lr=1e-6
)

# Train the model
history = model.fit(
    train_generator,
    validation_data=valid_generator,
    epochs=8,
    class_weight=class_weights,
    callbacks=[early_stop, reduce_lr],
)
# Fine-tuning: Re-train with a lower learning rate
model.compile(
    optimizer=tf.keras.optimizers.Adam(learning_rate=1e-5),
    loss="categorical_crossentropy",
    metrics=["accuracy"],
)

history_fine = model.fit(
    train_generator,
    validation_data=valid_generator,
    epochs=5,
    class_weight=class_weights,
```

```
    callbacks=[early_stop, reduce_lr],
)
# Evaluate the model
loss, accuracy = model.evaluate(valid_generator)
print(f"Validation Accuracy: {accuracy  100:.2f}%")
from sklearn.metrics import confusion_matrix, classification_report
import numpy as np

# Predict on the validation data
y_pred = model.predict(valid_generator)
y_pred_classes = np.argmax(y_pred, axis=1)  # Convert predictions to class labels

# Access true labels from y_test
y_true_classes = np.argmax(y_test, axis=1)  # True class labels

# Print classification report
print("Classification Report:")
print(classification_report(y_true_classes, y_pred_classes, target_names=enc.classes_))  # Use enc.classes_
for class names
# Plot training and validation accuracy
plt.plot(history_fine.history["accuracy"], label="Train Accuracy")
plt.plot(history_fine.history["val_accuracy"], label="Validation Accuracy")
plt.title("MobileNet Accuracy curves")
plt.xlabel("Epochs")
plt.ylabel("Accuracy")
plt.legend()
```

```
plt.show()

# Loss Plot

plt.plot(history_fine.history['loss'], label='Train Loss')

plt.plot(history_fine.history['val_loss'], label='Validation Loss')

plt.title('Loss Curve - MobileNet')

plt.xlabel('Epochs')

plt.ylabel('Loss')

plt.legend()

plt.show()

from sklearn.metrics import confusion_matrix, classification_report

import seaborn as sns


# Predict on the validation data

y_pred = model.predict(valid_generator)

y_pred_classes = np.argmax(y_pred, axis=1)  # Convert predictions to class labels


# Access true labels from y_test

y_true_classes = np.argmax(y_test, axis=1)  # True class labels


# Compute confusion matrix

conf_matrix = confusion_matrix(y_true_classes, y_pred_classes)


# Plot confusion matrix

plt.figure(figsize=(8, 6))

sns.heatmap(

    conf_matrix,
```

```
    annot=True,
    fmt="d",
    cmap="Blues",
    xticklabels=enc.classes_,  # Use enc.classes_ for class names
    yticklabels=enc.classes_,  # Use enc.classes_ for class names
)
plt.xlabel("Predicted Label")
plt.ylabel("True Label")
plt.title("Confusion Matrix - MobileNet")
plt.show()
i = 6
pred = np.argmax(model.predict(np.array([X_test[i]]))[0])
act = np.argmax(y_test[i])
print("Predicted class: {}".format(enc.classes_[pred]))
print("Actual class: {}".format(enc.classes_[act]))
plt.imshow(X_test[i])
```

CERTIFICATES OF CONFERENCE

A Research paper entitled “Real Time Traffic Optimization using Reinforcement Learning-based Adaptive Traffic Light Control” authored by Mr. P. Bhasha, Puthramaddi Nandini, U Vijay Kumar, V. S. Mustak Moulana, Revuri Dharmik is accepted for presentation in the “3rd International Conference on Data Analytics, Smart Computing and Networks (IDASCN -2024)” Organised by Department of Data Science & Computer Applications, Mohan Babu University (Erstwhile Sree Vidyanikethan Engineering College (Autonomous)) Tirupati-517102 during 24 - 26 October, 2024. The paper will be further indexed by Scopus database.

3rd International Conference on

DATA ANALYTICS, SMART COMPUTING AND NETWORKS

IDASCN'2024

MBU

Organized by Department of Computer Applications/Data Science

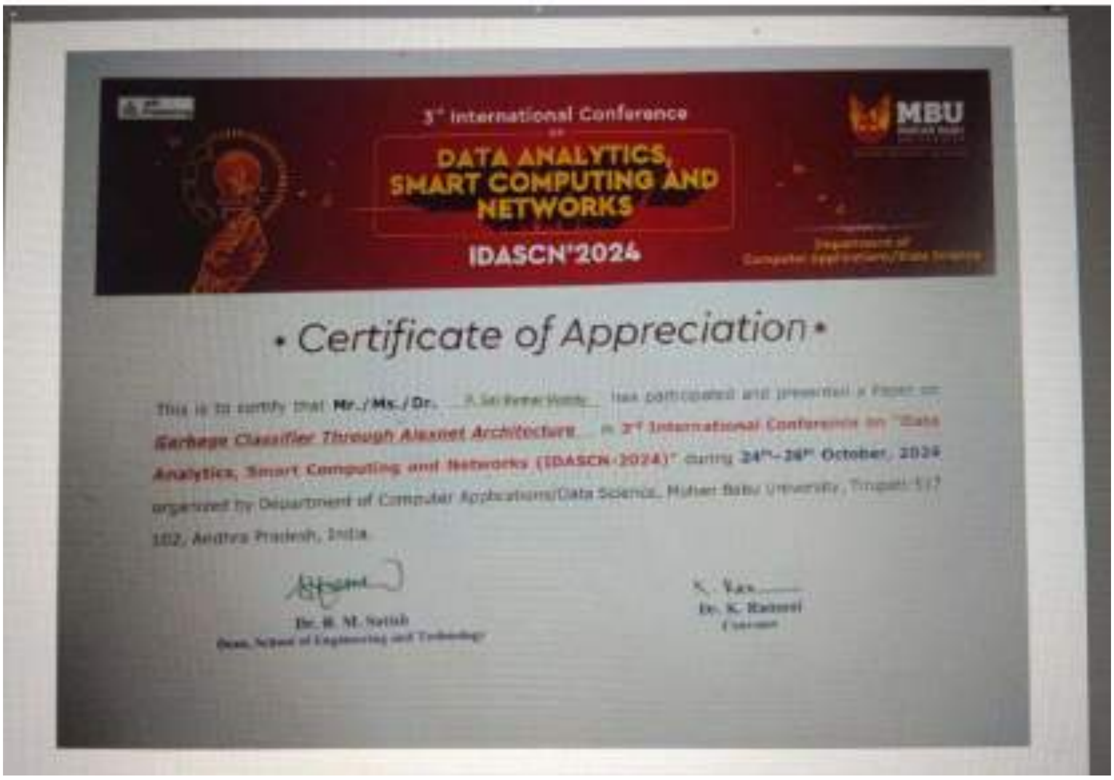
• Certificate of Appreciation •

This is to certify that Mr./Ms./Dr. S. K. Sameera has participated and presented a Paper on Garbage Classifier Through Alexnet Architecture in 3rd International Conference on "Data Analytics, Smart Computing and Networks (IDASCN-2024)" during 24th–26th October, 2024 organized by Department of Computer Applications/Data Science, Mohan Babu University, Tirupati-517 102, Andhra Pradesh, India.

Dr. B. M. Satish
Dean, School of Engineering and Technology

Dr. K. Ramani
Convener


3rd International Conference on
DATA ANALYTICS, SMART COMPUTING AND NETWORKS
IDASCN'2024
MBU
Department of Computer Applications/Data Science
• Certificate of Appreciation •
This is to certify that Mr./Ms./Dr. P. Charan Raj has participated and presented a Paper on Garbage Classifier Through Alexnet Architecture in 3rd International Conference on "Data Analytics, Smart Computing and Networks (IDASCN-2024)" during 24th-26th October, 2024 organized by Department of Computer Applications/Data Science, Mohan Babu University, Tirupati-517 102, Andhra Pradesh, India.
Dr. B. M. Satish
Dean, School of Engineering and Technology
Dr. K. Ramani
Convener


3rd International Conference on
DATA ANALYTICS, SMART COMPUTING AND NETWORKS
IDASCN'2024
MBU
• Certificate of Appreciation •
Garbage Classifier Through Alexnet Architecture
Analytics, Smart Computing and Networks (IDASCN-2024)" during 24th-26th October, 2024
102, Andhra Pradesh, India.
Dr. B. M. Satish
Dr. K. Ramani

COLLEGE VISION & MISSION

VISION

To be one of the Nation's premier Engineering Colleges by achieving the highest order of excellence in Teaching and Research.

MISSION

Through multidimensional excellence, we value intellectual curiosity, pursuit of knowledge building and dissemination, academic freedom and integrity to enable the students to realize their potential. We promote technical mastery of Progressive Technologies, understanding their ramifications in the future society and nurture the next generation of skilled professionals to compete in an increasingly complex world, which requires practical and critical understanding of all aspects.

DEPARTMENT OF COMPUTER SCIENCE AND SYSTEMS ENGINEERING

VISION & MISSION

VISION

To become a centre of excellence in Computer Sciences and Systems Engineering through teaching, training, research and innovation to create quality engineering professionals who can solve the growing complex problems of the society.

MISSION

- Established with the cause of development of technical education in advanced computer sciences and engineering with applications to systems there by serving the society and nation.
- Transfer of Knowledge through contemporary curriculum and fostering faculty and student development.

- Create keen interest for research and innovation among students and faculty by understanding the needs of the society and industry.
- Skill development among diversity of students in technical domains and profession for development of systems and processes to meet the demands of the industry and research.
- Imbibing values and ethics in students for prospective and promising engineering profession and develop a sense of respect for all.

PROGRAM EDUCATIONAL OBJECTIVES (PEOs)

After few years of graduation, the graduates of B. Tech (CSSE) will:

1. Demonstrate competencies in the Computer Science domain and Management with an ability to comprehend, analyze, design and create software systems for pursuing advanced studies in the areas of interest.
2. Evolve as entrepreneurs or be employed by acquiring required skill sets for developing computer systems and solutions in multi-disciplinary areas.
3. Exhibit progression and professional skill development in Computer programming and systems development with ethical attitude through life-long learning.

PROGRAM SPECIFIC OUTCOMES (PSOs)

On successful completion of the Program, the graduates of B. Tech (CSSE) program will be able to:

PSO1 Employ Systems Approach to model the solutions for real life problems, design and develop software systems by applying Modern Tools.

PSO2 Develop solutions using novel algorithms in High Performance Computing and Data Science.

PSO3 Use emerging technologies for providing security and privacy to design, deploy and manage network systems.

PROGRAM OUTCOMES (POs)

1. Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems (**Engineering knowledge**).

2. Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences (**Problem analysis**).

3. Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations (**Design/development of solutions**).

4. Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions (**Conduct investigations of complex problems**).

5. Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations (**Modern tool usage**).

6. Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice (**The engineer and society**).

7. Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development (**Environment and sustainability**).

8. Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice (**Ethics**).

9. Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings (**Individual and team work**).

10. Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write

effective reports and design documentation, make effective presentations, and give and receive clear instructions (**Communication**).

11. Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments (**Project management and finance**).

12. Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change (**Life-long learning**).

COURSE OUTCOMES (COs)

After successful completion of this course, the students will be able to:

CO1. Create/Design algorithms and software to solve complex Computer Science and allied problems using appropriate tools and techniques following relevant standards, codes, policies, regulations and latest developments.

CO2. Consider Society, Health, Safety, Environment, Sustainability, Economics and project management in solving complex Computer Science and allied problems.

CO3. Perform individually or in a team besides communicating effectively in written, oral and graphical forms on Computer Science and allied systems or processes.

Mapping of Course Outcomes with POs and PSOs:

Course Outcomes	Program Outcomes												Program Specific Outcomes		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	3	3	3	3	-	-	3	-	-	-	3	3	3	3
CO2	-	-	-	-	-	3	3	-	-	-	3	-	3	3	3
CO3	-	-	-	-	-	-	-	-	3	3	-	-	3	3	3
Average	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
Level of correlation of the course	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3