REPUBLIC OF TÜRKİYE

ALTINBAŞ UNIVERSITY

Institute of Graduate Studies

Information Technologies

TOWARDS CLEANER ENVIRONMENTS: A STUDY OF DEEP LEARNING MODELS FOR PLASTIC BOTTLE DETECTION

Huda Mahmood Hassoon ALTHABETI

Master's Thesis

Supervisor

Assoc. Prof. Dr. Sefer KURNAZ

İstanbul, 2025

TOWARDS CLEANER ENVIRONMENTS: A STUDY OF DEEP LEARNING MODELS FOR PLASTIC BOTTLE DETECTION

Huda Mahmood Hassoon ALTHABETI

Information Technologies

Master's Thesis

ALTINBAŞ UNIVERSITY

2025

The thesis titled TOWARDS CLEANER ENVIRONMENTS: A STUDY OF DEEP LEARNING MODELS FOR PLASTIC BOTTLE DETECTION prepared by HUDA MAHMOOD HASSOON ALTHABTI and submitted on 13/06/2025 has been **accepted unanimously** for the degree of Master of Science in Information Technologies.

____

Assoc.Prof. Dr. Sefer KURNAZ

Supervisor

Thesis Defense Committee Members:

Assoc. Prof. Dr. Sefer KURNAZ	Department of Computer Engineering, Altınbaş University	____________________
Asst. Prof. Dr. Hameed Mutlag Farhan FARHAN	Department of Computer Engineering, Altınbaş University	____________________
Asst. Prof. Dr. Serdar KARGIN	Department of Biomedical Engineering, İstanbul Arel University	____________________

I hereby declare that this thesis meets all format and submission requirements of a Master’s thesis.

I so certify that all of the data and material included in this graduation project was gathered in complete compliance with academic regulations and ethical standards. As required by the aforementioned rules and behavior, I also declare that all references listed in the Reference List have been referenced in the text, and vice versa, and that all unoriginal materials and conclusions have been cited in the text.

Huda Mahmood Hassoon ALTHABETI

Signature

DEDICATION

I dedication my thesis to:

University of Altinbas

My very respected supervisor

All my family members, especially my very dear mother and father .

ABSTRACT

TOWARDS CLEANER ENVIRONMENTS: A STUDY OF DEEP LEARNING MODELS FOR PLASTIC BOTTLE DETECTION

ALTHABETI, Huda Mahmood Hassoon

M.Sc., Information Technologies, Altınbaş University,

Supervisor: Assoc. Prof. Dr. Sefer KURNAZ

Date: 06/2025

Page: 68

The widespread use of plastic bottles in our daily lives is contributing to significant environmental issues, particularly in marine ecosystems. A substantial quantity of plastic bottles is being carried back to the mainland from the sea by waves, often becoming trapped in coastal areas. The detrimental impact of plastic waste, including plastic bottles, on coastal ecology is a matter of concern. Thankfully, artificial intelligence (AI) has found diverse applications in various sectors, including environmental initiatives, offering promising solutions to combat these environmental challenges effectively. This report aims to provide a classification of bottle plastic using data images in various situations. We preprocess with the dataset and we apply different artificial intelligence algorithms to perform this classification. The CNN model achieved the highest ACC, reaching an impressive 99%.

Keywords: Plastic Bottles, CNN, VGG16, MobilenNet, Xception, PSO_WaOA.

TABLE OF CONTENTS

Pages

LIST OF TABLES

Pages

LIST OF FIGURES

Pages

ABBREVIATIONS

AI	:	Artificial Intelligence
DL	:	Deep Learning
ML	:	Machine Learning
CNN	:	Convolutional Neural Network
VGG	:	Visual Geometry Group
ACC	:	Accuracy
PREC	:	Precision
REC	:	Recall
F1-S		F1-Score

1. INTRODUCTION

1.1 BACKGROUND AND MOTIVATION

Battle Tracker is an innovative and cutting-edge solution designed to revolutionize the way plastic bottle detection is performed in outdoor environments. Employing advanced image annotation technology, this system provides state-of-the-art ACC in crowded and heterogeneous scenes filled with plastic bottles. Its advanced algorithms carefully process visual data, tagging images to the pixel, and rapidly searching for plastic bottle in the outdoor environment. Battle Tracker comes from his greater vision to combat pollution and closely work with environmental monitoring bodies, waste management officials, and conservationists by seamlessly integrating into existing frameworks to efficiently track plastic bottle pollutants and enable timely intervention and sustainable waste management strategies connected with all ongoing networks. Battle Tracker takes plastic bottle litter to the next level with unparalleled real-time visibility into incidence trends and locations.

In addition, plastic waste has become one of the most pressing global environmental problems, endangering ecosystems and habitats around the world. One of the ecosystems most impacted by plastic pollution is the marine ecosystem [1]; once in the oceans, it poses widespread ecological, aesthetic and economic hazards due to plastic debris [2]. Scale of Plastic Pollution Over 300 million metric tons produced per year, 8 million metric tons per year go into the ocean 3. Plastic bottles, the main plastic used in households, play a major role in this crisis, since almost all bottled water is packaged in plastic [5]. According to a report in 2018 there are over 600 billion bottles per year [6] and though only 47% of them are being collected. Uncollected plastic bottles disperse into water bodies, soil, and sediment environments, typically flowing through rivers into the ocean. The cyclical movement of these uncaptured plastic water bottles, as they wash back and forth from the ocean into the ground, creates even more environmental harm for coastal zones.insert text here As a result, tracking and controlling plastic bottle waste in the environment are part of an important effort in reducing the impact on the environment.

1.2 RESEARCH MOTIVATION

This topic is born from the urgency to fight against the growing environmental impact of plastic bottle waste. However, plastic bottles are a pervasive, persistent threat in the outdoor space, and the consequences of the plastic bottle pollution crisis on our ecosystem, wildlife and human health are becoming more apparent every day. The absence of efficient and reliable techniques to detect and monitor plastic bottles in various outdoor environments makes it difficult to introduce targeted mitigation approaches. Battle Tracker aims to overcome this limitation with the help of new breakthrough technology in image annotation, artificial intelligence. Battle Tracker, by correctly recognizing and documenting those plastic bottles in the midst of complex microscopical backgrounds, aims to provide real-time data insights for environmentalists, conservationists, and waste management authorities. With the goal of reducing plastic bottles' detrimental effects on our planet's sensitive ecosystems and ocean environments, This study intends to support global efforts to conserve the environment and dispose of garbage in a sustainable manner.

1.3 PROBLEM STATEMENT

This thesis depicts a problem statement guided by this increasing environmental impact of plastic bottle pollution in outdoor environments. The ongoing extensive usage of plastic bottles has resulted in a threat to life and environment, especially in aquatic ecosystem. Despite many attempts to mitigate this problem, accurately detecting and tracking plastic bottles across a wide range of outdoor environments is currently a daunting task. Plastic bottle waste detection and monitoring are often not satisfactory in terms of PREC and efficiency with existing methods, which hinders tasking solutions supporting waste management and conservation. This has resulted in the urgent need to address this issue, as every year an enormous amount of plastic waste is produced, of which a large part ends up in our water bodies, soil, and coastal areas, wreaking irreversible damage to the environment. Plastic pollution has created a global environmental issue in recent years, and via my study, I hope to eradicate this problem by creating the innovative, high-tech "Battle Tracker" system. I use the latest image annotation technology and artificial intelligence to help outdoor plastic bottles get accurate detection. Battle Tracker aims to enable environmental stakeholders to make fast and informed decisions by providing real-time insights and data analytics, leading to a more sustainable approach to managing plastic bottle pollution and protecting the health of our planet.

1.4 RESEARCH OBJECTIVES

This body of work utilizes a multi-pronged approach to deal with the important environmental concern of plastic bottle pollution. The main objective is to design a robust and fast solution to detect and trace plastic bottles in various outdoor target application contexts, particularly in coastal and marine environment. This will have latest imaging annotation technology and artificial intelligence algorithms to reach high PREC and instant understanding. Another key objective is to integrate Battle Tracker seamlessly into existing environmental monitoring frameworks, enabling environmentalists, conservationists, and waste management authorities to access and analyze the collected data effortlessly. The research also seeks to enhance the awareness and understanding of the plastic bottle pollution problem, shedding light on its detrimental effects on ecosystems, aesthetics, and the economy. By fulfilling these goals, the study hopes to make a substantial contribution to conservation and sustainable waste management techniques. Ultimately mitigating the harmful impact of plastic bottle pollution on our planet's environment and marine life.

1.5 REPORT STRUCTURE

The rest of this research is splitted into several chapters:

Chapter 2: It discuss the waste management and the automated system for bottle plastics. Also it gives an overview of past research in the topic that aids in placing the current study into context.

Chapter 3: It demonstrates the suggested approach, presenting the different steps to achieving the classification of bottle plastic

Chapter 4: It shows the results of the different models applied for this topic and then there is a comparison between them.

Chapter 5: The conclusion encapsulates the principal discoveries and implications arising from this endeavor.

2. REVIEW OF THE LITERATURE

2.1 INTRODUCTION

This chapter examines the history of waste management and plastic bottle recycling, highlighting the pressing nature of the issue and its effects on the environment. We then go on to discuss automated systems and garbage sorting, presenting cutting-edge inventions and technologies that might revolutionize waste management. We also provide a thorough analysis of the current literature on plastic bottle recycling and trash management. Last but not least, we go through the various algorithms used in this domain.

2.2 PLASTIC BOTTLES

2.2.1 Waste Management and Recycling of Plastic Bottles

A very Important step towards fighting against Environmental degradation and pollution is efficient waste management, mainly recycling bottle plastic. Plastic bottles that do not decompose over time and their extensive use, it increasingly add to the world plastic waste problem. However, with some good waste management, especially with recycling, we can greatly reduce the environmental problem caused by plastic bottles that are thrown away. Properly organized recycling programs allow communities to divert a substantial volume of plastic bottles from landfills and incineration, thus saving valuable resources and preventing greenhouse gas emissions. Additionally, by recycling plastic bottles, we can help reduce the consumption of new plastic production, which reduces pressure on natural resources and ecosystems. Public awareness and involvement in recycling projects will be the foundation of the success of sustainable waste management activities. Together, by ensuring we recycle plastic bottles, We can safeguard the health of our world and build a cleaner, greener future. Pimback.

PW accounts for approximately12% of the world wide generated total municipal solid waste, and generally only 15% of that gets recycled. And an alarming 40% of this plastic waste is mismanaged, entering the environment [7]. The serious and widespread consequences of plastic pollution affect land, water and air quality, and biodiversity and climate change. Plastic waste takes hundreds of years to break down, leaching toxic chemicals and microplastics into the environment that can endanger wildlife and enter food chains [8]. The upstream procedures involved in the manufacture, transportation, and degradation of plastics also contribute to greenhouse gas emissions [9]. These statistics highlight the importance of waste management and the urgency of addressing the plastic pollution issue and its consequences for the ecosystem.

Figure 2.1: Bottles Recycling

2.2.2 Automated Systems and Waste Sorting

Waste sorting is the process of separating materials according to the kind of waste created, while automated systems facilitate this process with innovation and efficiency. The process of separating various waste products, such as plastics, paper, glass, and organic garbage, at the source or at recycling facilities is known as trash sorting. This is necessary to obtain recycling that is as efficient and therefore minimize the unavoidable residues that is sent to either landfills or inceneration; both of which will lead to bad environmental consequences. Automated systems utilizing cutting-edge technologies such as AI and robotics have also been integrated into the waste sorting process. Advanced systems that can recognize and sort numerous waste items simultaneously and with a great degree of accuracy and speed, enhancing overall sorting ACC and operational effectiveness. These systems lighten the load on human labour as well as increase recycling rates and resource recovery by automating sorting tasks. Waste sorting and automated systems are imperative for designing a cleaner, environmentally friendly future as we create a real solution for sustainable waste management.

Recent advancements in automated waste sorting systems offer the potential for more effective identification of the two classes of PBs and subsequent separation processes. These systems utilize advanced technologies including AI, ML, computer vision, and sensor-based mechanisms, which have been proven to perform exceedingly well in waste sorting tasks, both in terms of efficiency and ACC [10]. By using computer vision algorithms and image processing techniques, these systems can reliably recognize and classify PBs based on visual characteristics including form, color, and texture. Another reason is that sensor-based technologies, such as near-infrared (NIR) spectroscopy, enable PB detection by differentiating plastic types according to their molecular characteristics.

An automated waste sorting unites the advantages of the traditional approaches. They have faster sorting speeds, higher ACC, and the ability to process more waste. Moreover, automated systems may employ a variety of technology to sense contaminants and separate plastic components by type, even to the level of identifying specific laser values of PBs. Such advances can help significantly enhance the speed and accuracy of PB sorting, resulting in improved recycling rates and decreased contamination [11]. Lastly, when combined with all of the automated trash sorting technologies mentioned above, they are essential to environmentally friendly garbage management.

2.3 LITERATURE REVIEW

A Smart Waste Management system that uses automated machine learning algorithms to recognize when a recycling tank is empty based on signal measurements is described in the study [12]. The problem is framed as a binary classification task, with the over-representation problem to solve being a realistic case of emptying not happening in most events. Through analyzing several data-driven methodologies, the performance of the existing handcrafted model, its adapted counterparts, and traditional ML algorithms are assessed. The performance of the current handmade model, its modified equivalents, and conventional ML techniques are evaluated by examining a number of data-driven approaches. The classification ACC and REC of the manually constructed model are significantly improved by using machine learning approaches, achieving an astounding 99.1% and 98.2%, respectively, as opposed to the initial values of 86.8% and 47.9%. A Random Forest classifier and a feature set based on the filling level at various time periods are used in the top-performing solution. In addition, the improved approach outperforms the manually constructed baseline model in terms of forecast quality for recycling container emptying time prediction. This study demonstrates how automated machine learning may be used to solve real-world problems in smart waste management systems, resulting in waste management procedures that are more precise and effective.

In their paper [13], the authors tackle the urgent issue of water pollution caused by the extensive dumping of plastic and waste into oceans, posing a significant threat to the aquatic ecosystem. To support research in this area, they present Aqua Trash, a new dataset built on top of the TACO dataset. By utilizing DL techniques, the authors introduce AquaVision, an object detection model built around the AquaTrash dataset. This model's mean Average PREC (mAP) of 0.8148 makes it ideal for identifying and categorizing a variety of contaminants and hazardous waste in seas and on coasts.

The proposed method classifies waste objects with accurate localization information, allowing for targeted cleaning in water bodies. AquaVision helps to protect the aquatic ecosystem which in the long run, can help to protect our earth from the adverse effects of contaminated water caused by plastics and waste.

The authors in another paper [14] have proposed an image-based waste classification system which classifies waste as dry waste or wet waste. The primary objective is to facilitate efficient waste management by enabling the separation of different waste types for appropriate treatment. By adhering to waste segregation norms, substantial cost savings can be achieved, allowing budget allocation towards effective waste treatment. Because of the system's user-friendly design, municipal organizations may upload waste bin photos to the platform for analysis. Based on the supplied photos, machine learning techniques are used to identify the garbage's contents. Putting this system in place might provide insightful information on how garbage is disposed of in various places, leading to awareness campaigns and improvements in waste disposal practices. The trash categorization system's ultimate goal is to help create a cleaner environment and more sustainable waste management techniques.

This work [15] focuses on the use of DL, mainly CNNs, in image processing, especially for image analysis problems related to remote sensing. The contributed work's authors provide an unsupervised method that may be used to the detection of illicit trash dumps in the Saint Louis area of Senegal, West Africa, using unmanned aerial vehicle (UAV) photos. Due to the great spatial resolution and detail of UAV photos, this is a difficult undertaking that calls for appropriate automated analysis techniques. The proposed way is to divide the image into four regions and resizing the divided regions for feeding them as input to the CNN, and labeling the segments of selected interest. The model extracts highly descriptive features from the datasets using the Single Shot Detector (SSD) technique. These results indicate that the model finds the relevant areas, but in some cases it struggles to discern well in the areas where there is sparse or imprecise ground truth data. However, this study highlights the considerable promise of DL and CNNs in successfully tackling intricate image analysis challenges, especially in the field of remote sensing.

Article [16] has addressed many challenges generated by the waste management in terms of large and heterogeneous waste production types. In the theoretical framework, CNN is used in conjunction with ORB for object control and YOLO for object detection. The new system is scalable and efficient and can be applied to larger waste largeScale as a cost-effective way to manage waste by concentrating it at the district level. Evidently, by applying this model, we can have a substantial increase in our waste recycling rate which if achieved will have a lot reversal impact on the society. The roadmap to this framework would serve as the backbone for setting up a full-fledged waste management system based on the 3R principles (Reuse, Reduce, Recycle) for moving towards Sustainable Waste Management.

2.4 MACHINE LEARNING

ML[17] is a quickly developing area of artificial intelligence that uses algorithms to let computers learn and improve their performance over time. A type of artificial intelligence called machine learning (ML) enables computers to learn from data and gradually enhance their performance without explicit programming.. It means more like self learning from the data after data and optimizing its own performance to get better results.

This can be addressed using ML techniques as they can help to use features from the screen and make sense of the pictures collected with the aid of cameras. Machine Learning (ML) systems automate the sorting process which leads to a lowered dependency on manual labour, ultimately yielding cost savings and enhanced efficiency. Moreover, the objective and consistent operations of ML algorithms reduce variability and subjectivity, allowing for more consistent and accurate sorting for waste. The successful application of ML to waste sorting, however, is contingent on addressing multiple important factors. By collecting and labeling appropriate representative samples of waste materials, such as plastic bottles (PBs), we provide sufficient training data for the model to be effective. Training ML models on larger datasets and using more complex algorithms will also overall increase those computational resources (in terms of time and memory) that you will require to train and deploy these ML models. A further significant consideration is the integration of ML algorithms with existing waste sorting infrastructure and systems. ML can be used in conjunction with automated sorting machines or sensor-based technologies to augment their sorting capabilities. This integration requires careful coordination and calibration to operate smoothly and provide good performance [5].

Machine learning serves as a crucial foundation for crafting sophisticated applications such as: "Battle Tracker: Annotated Images for Accurate Detection of Plastic Bottles in Outdoor Settings." It leverages ML algorithms to detect and identify plastic bottles in challenging outdoor settings. Utilizing well labeled images, the ML model is trained to identify particular visual features related to plastic bottles, thus ensuring accurate detection in adverse environments. Overall, plastic bottle detection with ML based on annotated images has improved efficiency and ACC, contributing to combating plastic bottle pollution by empowering environmentalists and waste management authorities. By adding value and changing the way of disposing of organic waste, Battle Tracker is an astonishing step in improving waste management and sustaining the environment, showcasing that ML can be the game changer in different environmental crises.

2.5 DEEP LEARNING

DL[18] is one of the ML subsets that has astounded the AI community. To evaluate and learn from massive data sets, it makes use of multi-layered artificial neural networks. These networks use the forward and backward propagation methods to automatically identify intricate patterns and extract higher-level features from difficult input. DL has emerged as the cutting edge in a number of applications, surpassing traditional machine learning techniques in terms of efficacy and accuracy in fields including speech recognition, computer vision, and natural language processing.

Transfer learning is a potent component of DL that has drawn a lot of attention. This makes it easier to adapt and fine-tune previously trained models which were created on sizable datasets for certain tasks to new, related tasks with less labeled data. It allows faster and better training on new datasets since now the model can use the previously acquired knowledge from other tasks. Transfer learning has remained particularly beneficial in cases were gathering data is resource-intensive or there is limited labeled data.

For "Battle Tracker: Annotated Images for Accurate Detection of Plastic Bottles in Outdoor Settings," changing the DL and transfer learning of plastic bottle detection and classification can result in high ACC. Utilizing annotated images and pre-trained models, the system has gained proficiency in most outdoor settings, successfully detecting plastic bottles in real-time. It holds national importance for waste management, environmental preservation, and reducing carbon footprints in many countries worldwide. The use of these cutting-edge machine learning methods demonstrates how cutting-edge research may solve pressing environmental problems and improve the planet.

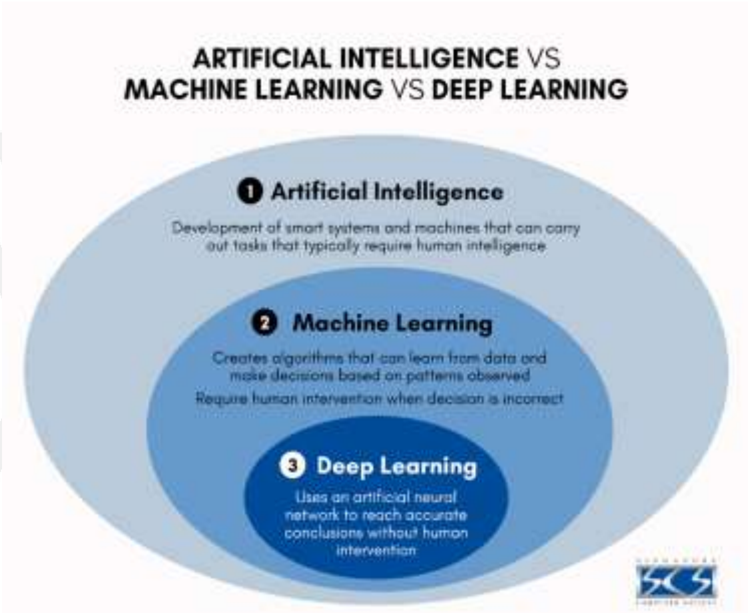

Figure 2.2: AI vs ML vs DL

2.5.1 CNN

Image classification tasks were relying on traditional neural networks such as Multilayer Perceptrons (MLP) in the early days. Yet the higher the image resolution, the more expensive it was for these networks and an excessive amount of parameters were needed to classify an image properly. CNNs[19] were proposed to address this limitation. CNNs have three-dimensional layers of neurons (width, height, and depth) where the traditional NN has a two-dimensional array. This structure enables CNNs to exploit the 3D shape of an image, making image classification more efficient. CNNs have therefore been preferred and extensively used in a variety of image categorization scenarios.

Convolutional, pooling, and fully connected are the three primary layers that make up a CNN's fundamental structure. The CNN can effectively analyze and extract significant information from pictures for a variety of applications, including segmentation, object identification, and image classification, thanks to the interaction of these three layers.

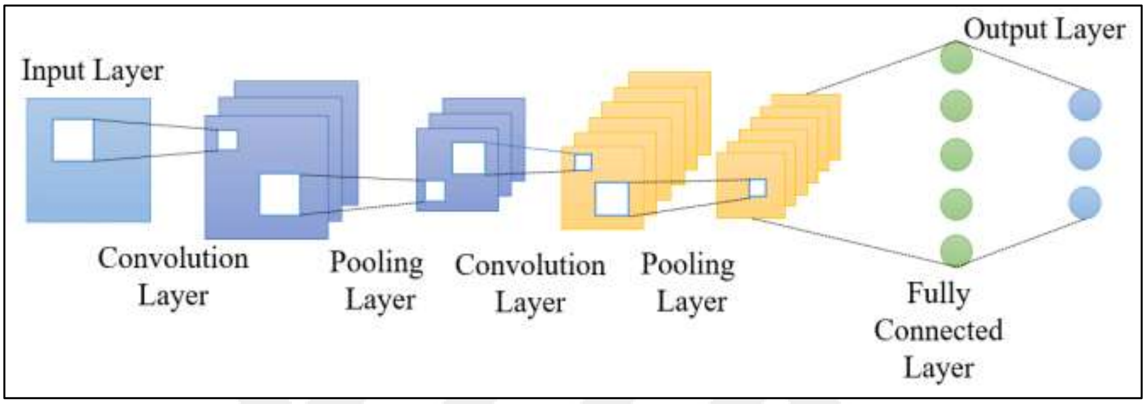

Figure 2.3: CNN Architecture.

2.5.2 VGG16

VGG-16 [21], denotes Visual Geometry Group 16, is a deep CNN structure characterized by its depth and extremely pleasing results in image recognition tasksVGG-16, also called VGG Net, was created by the University of Oxford's Visual Geometry Group and has 16 levels. Small 3x3 convolutional filters are used throughout the network, which is a distinguishing feature of VGG-16.With this, VGG-16 can build deep and complex representations of visual information, producing rich and hierarchical image features that capture complex shapes and patterns in different images. VGG-16 is a relatively simple architecture but has fared extremely well in many international competitions as it has been shown to classify images from a wide variety of classes with high accuracy. VGG-16 becomes one of standard benchmark in fields of DL because of its versatility and effectiveness and is the fundament of more sophisticated CNN architectures have been established.

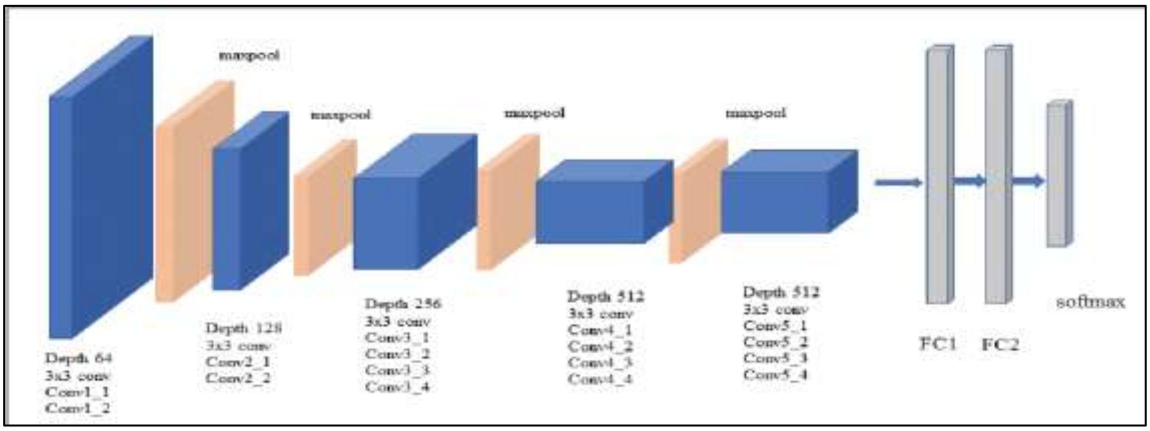

Figure 2.4: VGG16 Architecture.

2.5.3 MobileNet

MobileNet[22] is an advanced DL architecture which provides a solution to the challenge of deploying computationally budgeted CNNs on cost-sensitive devices. MobileNet is developed by Google researchers and uses a new architectural strategy called depthwise separable convolutions. The method breaks down conventional convolutions into two distinct processes: a pointwise convolution, which is a 1x1 convolution that concatenates the output channels, and a depthwise convolution, which applies a single filter to each input channel. Thus, MobileNet greatly decreases the model size and the number of calculations while achieving high ACC. MobileNet is particularly suited to tasks where resources are constrained as in the case with mobile phones, IOT devices and embedded systems. MobileNet has been a pioneering architecture for efficient DL empowering real-world deployment of advanced vision-based applications in edge devices.

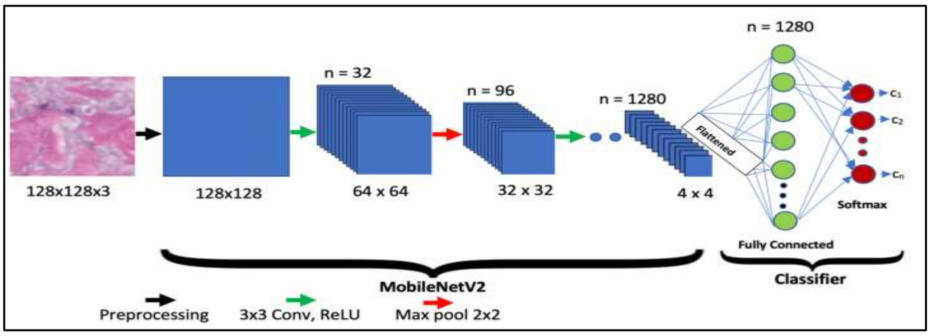

Figure 2.5: Example of MobileNet Architecture.

2.5.4 DenseNet

The novel and ground-breaking DenseNet[24] (Dense Convolutional Network) DL architecture has garnered significant interest in computer vision and other domains.

DenseNets were introduced by researchers at the University of Waterloo and they tackle several challenges such as improving information flow and feature reuse in deep networks. In contrast to typical CNNs, which carry information in a forwarding manner across the layers, DenseNet establishes dense connectivity among the several layers by connecting each layer to every layer amongst the same set of layers in a block. Information and gradients are more likely to spread throughout the network when there is such intense connection, increasing feature reuse and more efficient training. It allows to build deeper networks with fewer parameters and uses less memory. In addition, DenseNet also performs exceptionally well with relatively few training images, which is a boon for resource constrained situations. DenseNet's outstanding performance on image recognition tests has made it one of the most appealing architectures and a potent substitute that has spurred more study and made a substantial contribution to the development of sophisticated DL models.

2.5.5 Xception

The Xception-based[25] model is a deep learning(DL) architecture extending the Inception model by using depthwise separable convolutions for performance and efficiency. It is especially appropriate for challenging picture identification jobs as it learns to extract rich and significant characteristics from input data. As a result, the network can learn hierarchical representation from the input pictures thanks to the abundance of convolutional layers in the Xception model. An essential component of the suggested design, skip connections aid in resolving the vanishing gradient issue and enable effective training of very deep networks.

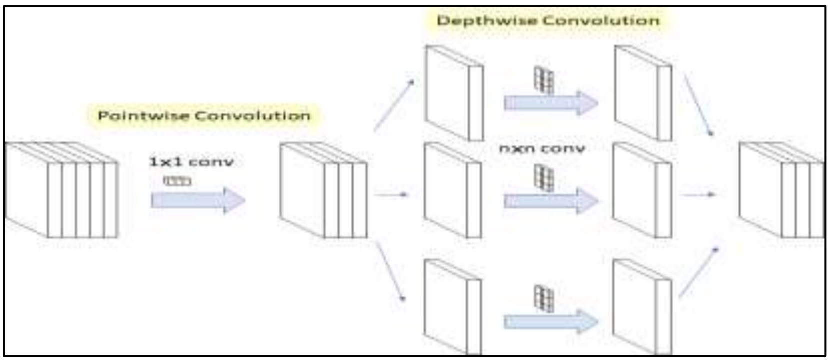

Figure 2.6: Xception Model.

2.6 METAHEURISTIC ALGORITHMS

In the realm of metaheuristic algorithms, the prefix 'meta-' denotes a level beyond or higher than simple heuristics. Generally, these algorithms outperform basic heuristics, striking a balance between local search and global exploration. Randomization is a common element employed to introduce diversity in the solutions generated. There isn't a single, accepted definition of heuristics and metaheuristics in the literature, despite the fact that metaheuristics are used extensively. Although some academics use the words interchangeably, the general consensus is that any stochastic algorithms that involve global exploration and randomization fall within the category of metaheuristics. The incorporation of randomization proves to be a valuable strategy for transitioning from local search to a broader exploration of the global scale. As a result, the majority of metaheuristic algorithms are applicable in global optimization and nonlinear modeling situations.

Through a methodical exploration of answers through trial and error within a suitable timescale, metaheuristics provide an effective technique to solving complicated issues. It is impractical to thoroughly search every possible combination or solution due to the complexity of the issues at hand. Rather, the objective is to find workable, acceptable solutions in a reasonable amount of time. Notably, there is no guarantee that the optimal answers will be found, and it is possible that an algorithm may not work as intended or that the factors that led to its success will not be known [37], [38]. The main goal is to create a useful and effective algorithm that continuously delivers excellent results and high-caliber solutions. While optimal solutions are not guaranteed, it is anticipated that among the generated quality solutions, some may approach optimality, though such optimality is not assured.

2.6.1 Energy Valley Optimizer

The Energy Valley Optimizer (EVO) [44] is a new metaheuristic algorithm that focuses on stability and various particle decay patterns, drawing inspiration from advanced physics concepts. The fundamental idea of EVO is based on the physics of particle decay, in which unstable particles release energy through decay or disintegration to produce lower-energy particles. This process is dictated by stability, which depends on the N/Z ratio of particles and interactions with other particles. EVO applications revolve about this phenomenon where particles move towards a stability band or “energy valley” to get to a stable state.

There is a multiple steps in the EVO algorithm. First, a portion of the universe is simulated by creating the space for solution candidates. The Enrichment Bound (EB) is subsequently defined to discriminate between neutron-rich and neutron-poor particles according to each particle's Neutron Enrichment Level (NEL). Particle fitness evaluations derived from objective functions used to categorize particles based on their stability levels (BS and WS).

The N/Z ratio is therefore over the critical limit when a particle's NEL is more than EB in the normal search loop, and alpha, beta, or gamma methods are used to mimic this download as it proceeds to decay. As stated above, mathematical imaging is used to update the particle positions in the algorithm while mimicking the physical emulation found in particle decay processes.

EVO employs beta decay as its position updating mechanism, particularly when a particle's stability level is below its stability bound. This leads a guided movement along with molecules or candidates with the most effective stability level, which mimics the attitude of molecules to reach an equilibrium molecular state.

In addition, it models an enhancement process for the exploration and exploitation degree. To further explain, this is a second method for updating positions with beta decay through respective stability of the particle undergoing the update (this means that it will move towards particles with the best stability level but not only mates as a new position will surely move a particle neighbor to the level of stability).

When a particle's neutron enrichment level falls below the so-called enrichment bound, which shows a reduced N/Z ratio, the particle either experiences positron decay or electron capture and moves toward the stability band. It is reasonable to assume that a particle's stability will rise as its neutron enrichment decreases. This manifests itself as a stochastic/ random moving around in the search space (space of all possible solutions) for the particle.

At the conclusion of the main loop for the subsequent search cycle, the newly created position vectors for every particle are finally included into the existing population. The algorithm additionally includes a stopping condition that limits the amount of objective function evaluations or iterations and an extra flag that indicates which choice variables are out of bounds.

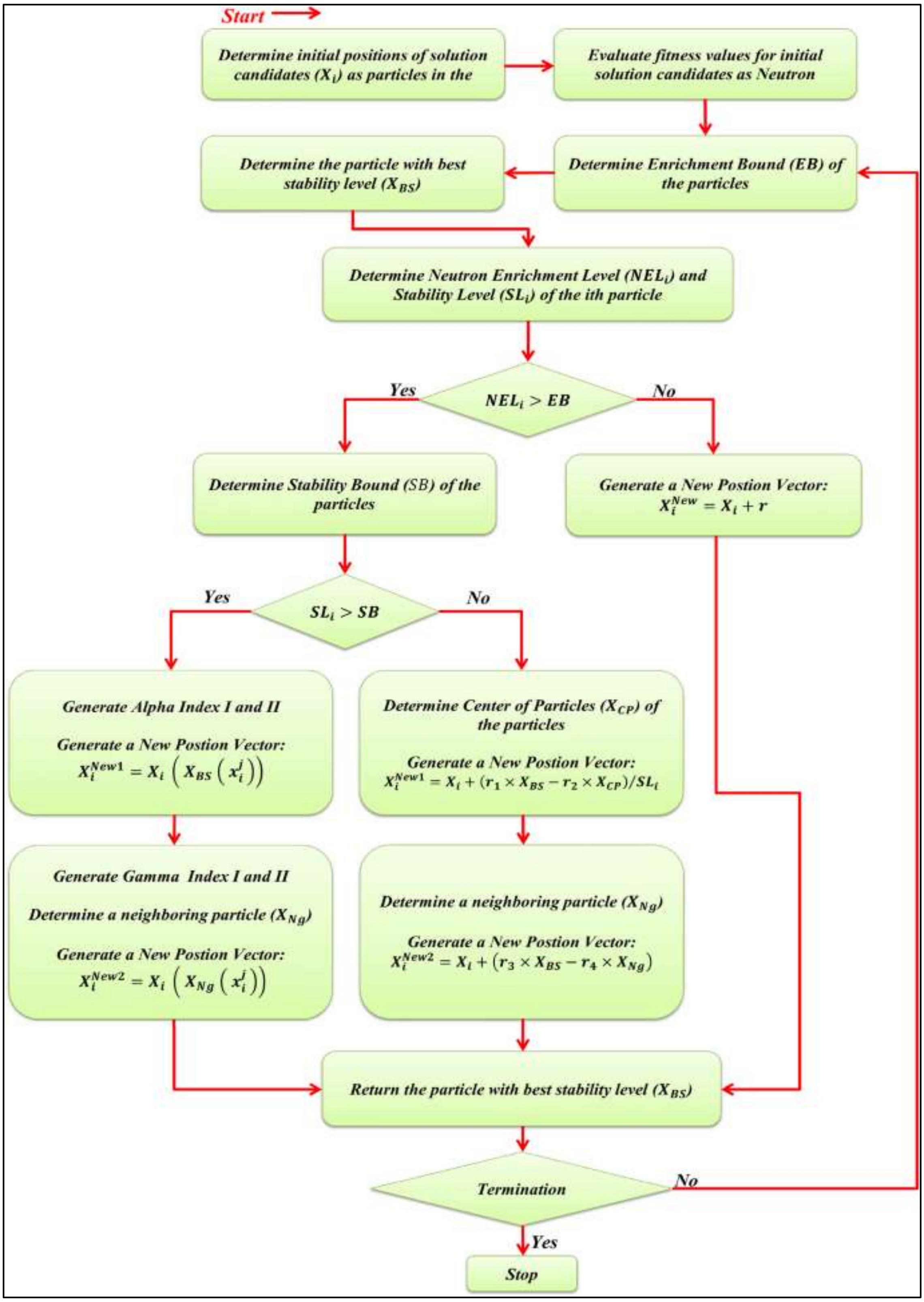

Figure 2.7: EVO Flowchart.

2.6.2 Whale Optimization Algorithm

A swarm intelligence-based optimization algorithm inspired by nature, the Whale Optimization Algorithm (WOA) mimics the humpback whale's hunting technique. WOA is similar to other algorithms in its research group like PSO and GWO, sharing a characteristic collective behavior. It is important to note that WOA has a special way of searching, which makes it a prominent and effective global optimizer, balancing the exploration and exploitation.

The complex bubble-net feeding behavior of humpback whales serves as the inspiration for the whale optimization algorithm (WOA) [39, 40, 41, 42, 43]. The technique depends on humpback whales' ability to locate their prey and effectively enclose it in a bubble net. Humpback whales are particularly known for swimming in a spiral motion upwards from about 15 meters down and blowing large and small bubbles in an upward direction. The bubbles climb in unison, whereas the first and last bubbles ascend simultaneously, thus forming a tubular or cylindrical net of bubbles. This net kind of looks like a giant spider web that catches a victim and leads it toward the center. In turn, the humpback whales align themselves in a vertical orientation inside the bubble ring, with mouths ajar, to swallow the entrapped prey. The knowledge-based model also uses an opportunistic approach based on the behavioral model to make more refined decisions for the different phases of a humpback whale's hunt, which include the prey seeking phase, spiral bubble-net feeding technique, and prey segmentation.

2.7 PARAMETERS AND HYPERPARAMETERS OF MODELS

For ML and DL models, parameters and hyperparameters [26] are two of the most critical elements determining the model's behaviour and performance. During the learning phase, the model learns these internal variables from the training data. In order to reduce the model's loss function and improve its predictions, they indicate the weights and biases of the neural networks' subsequent connections, which are modified collectively in subsequent iterations. The model architecture and the learning process are impacted by hyperparameters, which are external factors that determine the model's parameters prior to training. The model's external hyperparameters need to be adjusted for optimal performance. variables such as the network's number of layers, learning rate, number of neurons in each layer, batch size, etc. A crucial stage in the creation of a model is hyperparameter tweaking, which greatly influences the model's ACC and generalizability. Hence, both parameters and hyperparameters should be set correctly to develop effective and robust ML models.

2.7.1 Activation Function

An activation function is a vital part of artificial neural networks, which acts like a non-linear transformation to the output of neurons of every layer. In order to enable the model to learn and recognize intricate correlations and patterns in the data, one of its goals is to inject non-linearity into the network. One of the most popular activation functions in DL is the Rectified Linear Unit (ReLU), which maintains positive values constant while setting negative values to zero. Because it is easy to compute and effective, the Rectified Linear Unit (ReLU) is frequently utilized. It also helps to mitigate the vanishing gradient issue, which can cause training in deep networks to lag. Tanh and Sigmoid are additional activation functions., they are used in earlier days now they are not much used because they suffered from vanishing gradient problem and saturation issue. Researchers investigate and create novel activation functions for neurons in order to enhance the performance of DL models in a variety of complicated datasets.

2.7.2 Loss Function

A loss function [27], which is the model's performance metric and what is optimized, is a component in training DL models; The loss function, which calculates the discrepancy between the ground truth and the expected output, is optimized by the model during training. Usually, the loss function depends on the kind of job the model is completing, such as categorical cross-entropy in multi-class classification and mean squared error (MSE) in regression tasks.. Choosing an appropriate loss function is absolutely critical because it dictates how the model learns and how well it will be at predicting. An appropriate loss function helps to steer the model to correct predictions, while an inappropriate one may not only slow the learning process or converge to the wrong solution. As such, the thoughtful choice and tuning of the loss function is a key consideration in building successful, effective ML models.

2.7.3 Optimizer

When exploring the effective role of the optimizer in the training of ML and DL models, it essentially updates the model parameters to minimize the loss function and optimize model performance. Predicting the optimal model parameter values is the only thing that matters in order to improve predictions. Different optimizers must have been applied to achieve the optimization. Several well-known optimizers are Adam, RMSprop, Adagrad, Stochastic Gradient Descent (SGD), and others. They use techniques like momentum, adaptive learning rates, and other strategies to adjust the wights on the way. Different optimizers tend to give slightly unique results in terms of model performance, and a faster training time as well. Exploring how your model behaves with different optimizers and hyperparameter tuning for those optimizers is a critical move to enhance your ML Model and making sure your model performs well in the relevant task.

2.7.4 Epochs

The total number of times the training dataset runs through the model during training is known as the epochs [28]. An epoch is a whole training run of the complete dataset during which the model learning process modifies its parameters based on both the optimizer and the training data. Since it indicates how frequently we wish to learn from the data, the number of epochs is another crucial hyperparameter. A model that has too few epochs may underfit (fail to capture the patterns in the data) while one that has too many epochs may overfit (memorize the training data). Usually, a trial-and-error procedure is used to determine the ideal number of epochs, where we observe the behavior of validation metrics on separate validation data to ensure both good generalization and overall performance of our model.

2.8 CONCLUSION

Overall, the chapter examined the important topic of waste management and plastic bottle recycling, highlighting the environmental issues related to plastic pollution. By investigating options such as automated systems and waste sorting technologies, we demonstrated how advanced AI and ML solutions could be useful for quickly locating plastic bottles in outdoor locations. The literature review highlighted the importance of DL models in combating this issue and presented models like VGG-16, MobileNet, and DenseNet that have shown superior performance in image recognition tasks. Moreover, we discussed the importance of parameters and hyperparameters in tuning these models for better performance. Using these present-day exotic processes, we can develop an advanced approach for accurate detection and recycling of plastic bottles, aiding the cause to save the environment and a better tomorrow. With ongoing advancements in this field, the application of AI-powered technologies for waste management could potentially lead to a cleaner, greener final planet.

3. METHODOLOGY

3.1 INTRODUCTION

In this chapter, we will take a detailed tour on the foundational elements of our work with emphasis on the robust methodology used to solve the complex problem of bottle plastic classification. We start by highlighting the complex web of our thesis approach, followed by an elaboration of the conceptual framework of our exploration. We also give a thorough explanation of the data, including its amount, diversity, and composition, all of which are crucial building blocks for our models. Data preprocessing pipeline. At a deeper level, we will expose ourselves to several different steps that help us bring some of the raw data back into a state that it can be fed to our models. With our data ready to go, we make the foray into the world of data modeling, where we tap into a wide range of cutting-edge approaches to extract patterns from a very complex set of plastic bottle images. Last but not least, we discuss our thorough data evaluation protocols, moving through various open metrics to verify the effectiveness, efficiency, and resilience of our models.

3.2 THESIS APPROACH

The methodology used in this report is a systematic one that addresses the issue of accurately detecting plastic bottles in outdoor settings. The first step involves preprocessing the collected data to make sure their quality is high and are compatible with the detection algorithms. To optimize parameters in the detection model, which is essential for maintaining the improved performance in different environmental circumstances, a complex algorithm known as Energy Valley Optimization (EVO) is used.

The Whale Optimization Algorithm (WOA) is then used following the EVO stage, most likely to improve the model's hyperparameters by simulating humpback whale social behavior. Convolutional neural networks (CNNs) are created with a configured architecture to transport data via the input, convolutional, pooling, and dense layers to the output layer. The parameters adjusted by EVO and WOA are then integrated into the CNN.

Based on CNN, and different architectures including VGG-16, MobileNet, MobileNetV2, DenseNet-121, and Xception are the basis for architecture with special shapes and powers. These models are evaluated and compared to find the best architecture for accurate detection of plastic bottles in outdoor scenes. Then the main model is chosen based on all the metrics we mentioned, so we ensure as much accuracy and performance as possible.

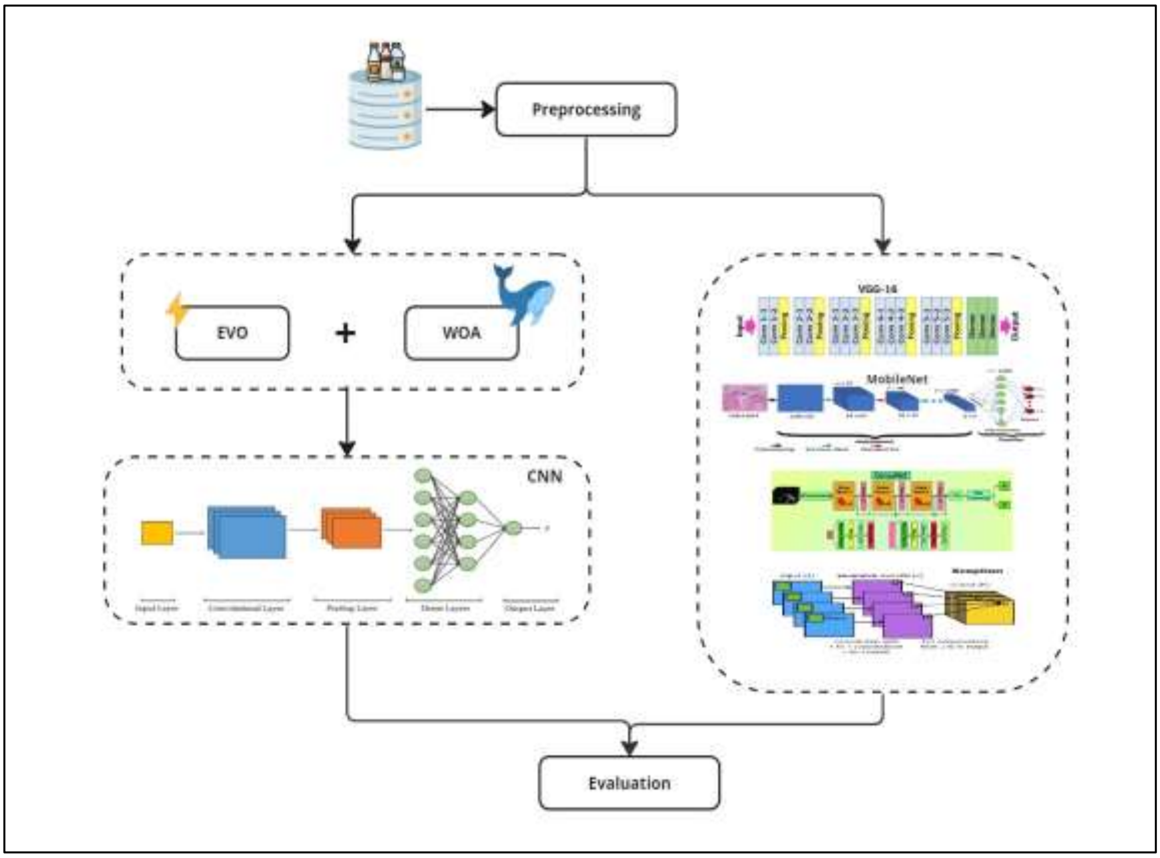

Figure 3.1: Methodology of Bottle Plastic Classification.

3.3 DATASET DESCRIPTION

The dataset used in this study includes around 8000 pictures of plastic bottles taken in different outdoor environments with bounding boxes annotated using the YOLO format. This dataset is used to train and validate object detection approaches on picking plastic bottles in an end-user setting. The images include a variety of backgrounds natural and urban and the bottles can be partially occluded or blend into their surroundings. The content of the dataset possesses such variabilityThis makes it a challenging yet doable test for object detection researchers and developers, thus the dataset will contribute towards their work. Due to this, the rich and relevant dataset will help tackle plastic waste and pollution more effectively and therefore will allocate a better model to detect plastic bottles in the outdoor environments.

Figure 3.2: An Overview of the Dataset.

3.4 DATA PREPROCESSING

This stage of the development of our plastic bottle detection model is known as data preprocessing [29]. First, the input images are processed to access their uniformity and compatibility. The data is encoded into JPEG format, trading a bit of image quality for not quite so much storage space. We also standardize the color spaceto RGB, resulting in the same representation of colours across all images.

Resize Images to a Common Resolution For Efficient Training The use of all images of the same size is important as all images fed into the model must be of the same size to avoid distortion.

Then, the dataset is split between two classes, plastic bottle, and non-plastic bottle. Separating the model from the object enables it to accurately differentiate what was the object and what was potential background clutter.

Since an imbalanced dataset can hurt the model's performance, we handle balancing so that each class is represented evenly. In this approach, the minority class is oversampled, or the majority class is undersampled such that a balanced dataset is created.

During training, we also shuffle the dataset to introduce randomness and avoid potential ordering effects. The data must be randomly given in order to guarantee that the model learns and generalizes to unseen data well.

Finally, To accurately assess the model's performance, we separated the dataset into three sections: the training set, the validation set, and the test set. This means that 80% of the data used to train the model is in the training set. Hyperparameters are optimized using the validation dataset to prevent overfitting, and tests are performed to provide a final evaluation and verify generalization.

These thorough data preprocessing steps also form the basis for the training and evaluation of our plastic bottle detection model, which is essential for effectively addressing the critical issue of plastic waste management in outdoor environments.

Figure 3.3: Examples of the Dataset.

In the figure, we mention the types of things found and precise the class; if it’s: "bottle-blue", "bottle-green", "bottle-dark", "bottle-milk", "bottle-transp", "bottle-multicolor", "bottle-yogurt", "bottle-oil", "cans", "juice-cardboard", "milk-cardboard", "detergent-color", "detergent-transparent", "detergent-box", "canister", "bottle-blue-full", "bottle-transp-full", "bottle-dark-full", "bottle-green-full", "bottle-multicolorv-full", "bottle-milk-full", "bottle-oil-full", "detergent-white", "bottle-blue5l", "bottle-blue5l-full", "glass-transp", "glass-dark", "glass-green" .

3.5 DATA MODELING

3.5.1 CNN

The CNN model tuned with Energy Valley Optimization (EVO) and Whale Optimization Algorithm (WOA) algorithms consists of a sequential stack of layers specifically arranged for binary image classification. It starts with an input layer, set to take in images of shape (224, 224, 3), which a number of convolutional layers get as input. Using (3, 3) kernels and ReLU activation functions, these convolutional layers are configured with an increasing number of filters, starting with 32 and going up to 64 and finally 128. Max pooling using a (2, 2) window decreases the spatial dimensions following each convolutional layer.

A flattening technique is used to convert the 2D feature maps into a 1D feature vector once the convoluted features have been pooled. The ReLU activation function is then used to send this vector across a dense layer of 128 neurons. The model's last layer consists of a single neuron with a sigmoid activation function that produces a probability indicating if a plastic bottle is present in the picture.

The model's parameters are refined through EVO, which likely mimics a natural process to avoid local minima and improve the learning path of the network. Further fine-tuning is done using the WOA, motivated by the hunting habits of humpback whales, with an emphasis on maximizing hyperparameters like batch size and neuron count.

Accuracy is tracked as the performance metric during the training phase, when the model is assembled using the Adam optimizer and binary cross-entropy loss. To make sure the model performs effectively when applied to fresh data, it is trained using a validation split.

Accuracy, loss, a thorough classification report, and a confusion matrix are used to characterize the model's performance after it has been tested on a test set. A heatmap is frequently used to illustrate the confusion matrix and show how well the model performs in categorizing the photos into the appropriate groups.

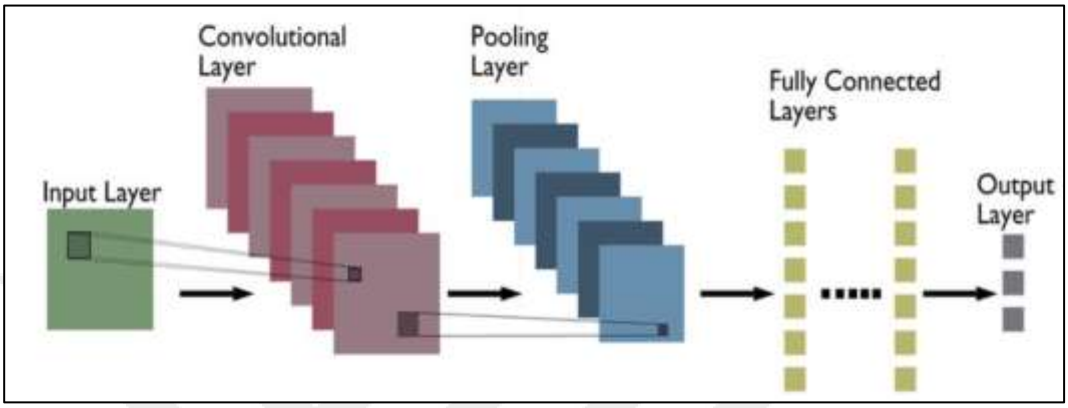

Figure 3.4: CNN Architecture.

3.5.2 VGG16

The transfer learning model presented here is designed to leverage the power of pre-trained VGG16, a widely used deep CNN architectureWithout having to train the entire model from start, this method allows the model to take advantage of the feature extraction skills acquired from a sizable dataset of natural photos (Imagenet). The VGG16 architecture is used to initialize the base_model. where the final fully connected layers (top layers) are excluded ('include_top=False') to allow for customization for our binary classification task.

The pre-trained weights in this model are guaranteed to stay constant throughout training since the base_model's pre-trained layers are frozen ('base_model.trainable=False'). This prevents overfitting and provides a stable starting point for further fine-tuning.

To convert the retrieved features into a one-dimensional vector, the Flatten layer is applied after the VGG16 base_model in the transfer learning model. A final dense layer with a single neuron activated by the sigmoid function is then added, creating a probability score that indicates the likelihood that the input image belongs to either the positive class (plastic bottle) or the negative class (non-plastic bottle). The first layer has 128 neurons activated by the ReLU function, which facilitates non-linearity.

The model is appropriate for binary classification tasks and is compiled using the Adam optimizer and binary cross-entropy loss. Next, using val_images and val_labels as validation data for 10 epochs and a batch_size of 32, the model is trained on the train_images and train_labels.

We chose Transfer Learning Model to detect plastic bottle outdoors for this task is we are leveraging VGG16 feature extraction advantage. Thus, by utilizing the pre-trained weights and having to only fine-tune the last few layers, the model can produce predictive accuracy with lower training time and less computational resources, which makes it most appealing for real-world applications.

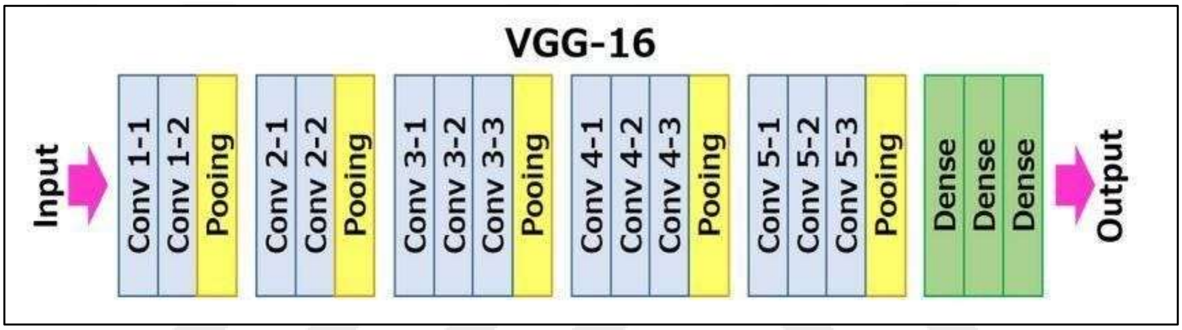

Figure 3.5: VGG16.

3.5.3 MobileNet

The transfer learning model based on MobileNetV2 suggested in this research is also an efficient solution for outdoor plastic bottle detection. MobileNetV2 is a shallow architecture of Deep CNN that is also appropriate for constrained environments. As our base_model, we choose the pre-trained MobileNetV2, which has been initialized with weights learned from Imagenet, and remove its eventual fully connected layers since we are performing binary classification.

The pre-trained layers in base_model are frozen, preventing overfitting as the number of data will be smaller, thus saving computational time. The model for transfer learning is a base_model of MobileNetV2, followed by a GlobalAveragePooling2D layer that reduces the feature dimensions effectively.

Two dense layers are then added, the first of which has 128 neurons with a ReLU activation function to introduce non-linearity. To ascertain whether or not the image depicts a plastic bottle, the last dense layer using a sigmoid function generates a single score. Binary cross-entropy is used as the loss function for the binary classification problem, and the Adam optimizer is used to create the model. Here, we validate the model using 10 epochs of val_images and val_labels after fitting it using train_images and train_labels.

MobileNetV2 Based Transfer Learning Model: The MobileNetV2 is an efficient, lightweight Deep Learning architecture designed for feature extraction, which achieves an excellent ACC while requiring a significantly low computational cost for real-life production.

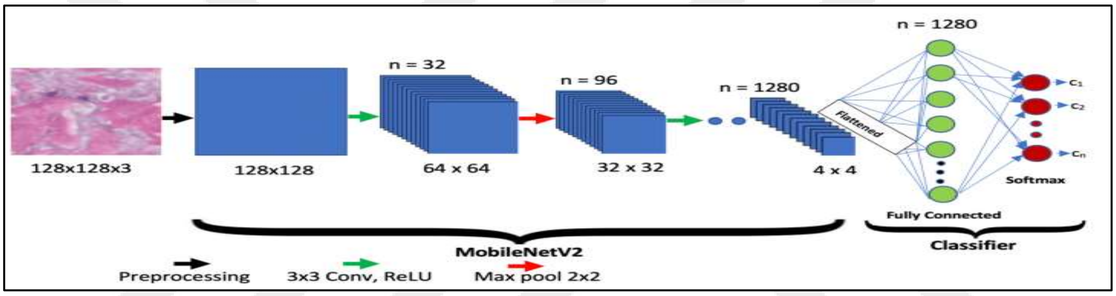

Figure 3.6: MobileNetV2 Architecture.

3.5.4 DenseNet

The model based on DenseNet-121 is an effective model for outdoor plastic bottle detection. It uses a pre-trained on Imagenet weights to extract features through a dense connection. Without the last fully connected layers, we modify the model for our binary classification problem.

The frozen pre-trained layers prevent overfitting. A GlobalAveragePooling2D layer is also included in the model to reduce spatial dimensions.

Two dense layers are then added; the first employs ReLU activation for non-linearity and has 128 neurons. The likelihood score for plastic and non-plastic bottles is predicted by the last dense layer, which has a sigmoid activation function.

The model is compiled using the Adam optimizer with binary cross entropy loss, and it is then trained on train_images and train_labels using val_images and val_labels for 10 epochs with a batch size of 32.

The denseNet-121-based model has high ACC and performance on plastic bottle detection in heterogeneous outdoor scenes, which is the next step towards practical applications.

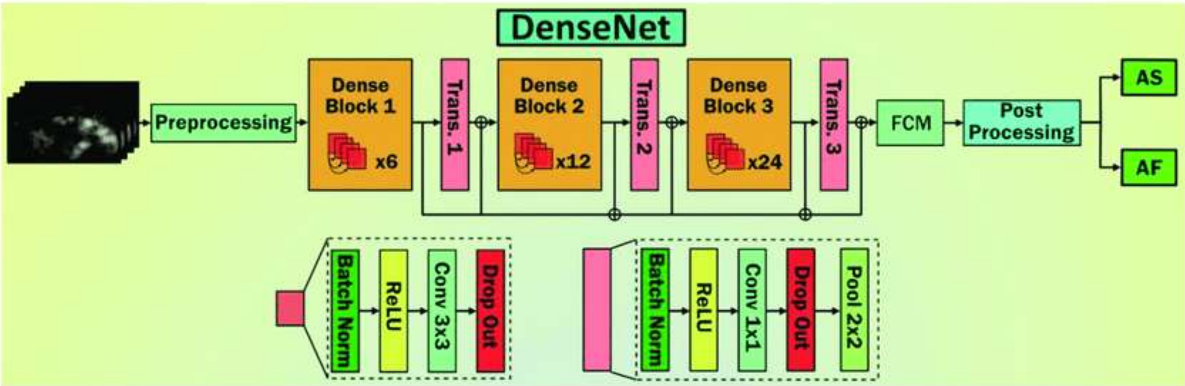

Figure 3.7: DensNet Model Architecture.

3.5.5 Xception Model

The Xception CNN serves as the backbone of the DL architecture that makes up the Xception-based model. The Inception model serves as the foundation for the Xception model, which employs depthwise separable convolutions to boost effectiveness and performance. The pre-trained Xception model, which was initialized with weights acquired from a sizable picture dataset, served as the basic model in this application.

The base model is fine-tuned for our specific task, where we aim to detect plastic bottles in outdoor settings. In order to do this, we employed a thick layer of size 128 with ReLU activation as the head, followed by a global average pooling layer that eliminated the spatial dimensions from our base model. The feature extractor here is a dense layer and also it contributes in reducing the overall model complexity.

The sigmoid function activates the last layer of the model, which is a dense layer with one unit in our binary classification problem (plastic or non-plastic bottle). Sigmoid activation outputs probabilities, hence we can interpret the output as the probability of an image containing plastic bottle.

The binary cross-entropy loss function, which measures the degree of discrepancy between predicted and true labels, was used for training when we assembled the Xception-based model using the Adam optimizer. During the training phase, we utilized a dataset of plastic bottle images annotated with bounding boxes in different outdoor environments. The training phase lasted through a total of 10 epochs, In contrast, the model's ability to generalize to new data was tested on a separate validation set. The performance indicators were calculated to assess how well the model performed in detecting plastic bottles in realistic scenarios, the ACC being one of them.

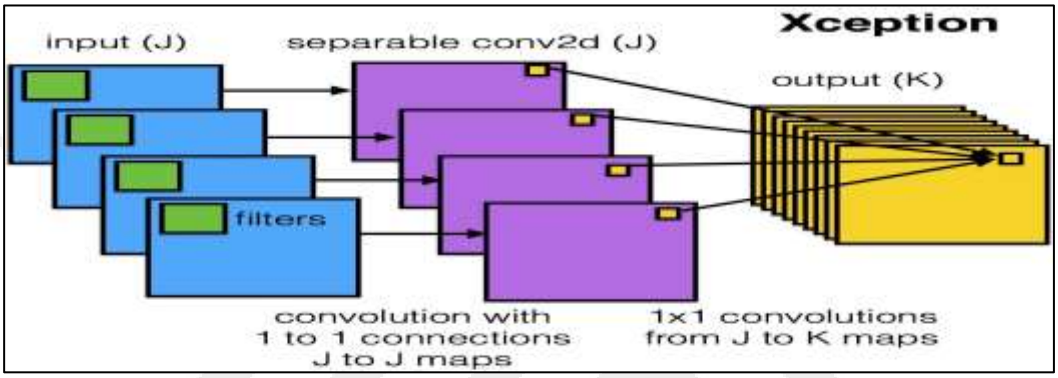

Figure 3.8: Xception Model Architecture.

3.6 EVALUATION METRICS

They are crucial instruments for assessing the effectiveness and performance of ML models. These metrics provide information about the model's performance in a particular job, such as regression or classification. Acc, Prec, Rec, F1-S and the confusion matrix are common classification evaluation metrics.

3.6.1 Classification Report

The accuracy, recall, F1 score, and other metrics of a classifier model are succinctly summarized in a classification report [30]. Giving important evaluation metrics like PREC, REC, F1-S, and ACC for every class in dataset.

a. Accuracy

One of the main metrics that machine learning uses to assess a model's interaction accuracy is ACC[31]. Additionally, it describes the proportion of correctly classified examples to all instances in the collection.

$$Accuracy = \frac{(TP+TN)}{(TP+TN+FP+FN)} \quad (3.1)$$

b. Precision

Out of all the cases the model predicts as positive, PREC[32] calculates the percentage of positively labeled events that it accurately predicts. A higher PREC value means the model has a low false positive count, making it reliable when it comes to classifying positives well.

$$Precision = \frac{TP}{(TP+FP)} \quad (3.2)$$

c. Recall

A crucial assessment parameter in machine learning for classification tasks is REC[33], often known as the sensitivity or true positive rate. It measures how well the model can distinguish positive examples from all of the dataset's real positive examples.

$$Recall = \frac{TP}{(TP+FN)} \quad (3.3)$$

d. F1-score

In ML, F1-S is an important evaluation metric for classification tasks, as it balances PREC and REC. It is their harmonic mean, which balances false positives with false negatives in a single number.

.

$$\mathrm{F1-Score} = 2 * \frac{(Precision*Recall)}{(Precision+Recall)} \quad (3.4)$$

3.6.2 Confusion Matrix

A particularly useful technique in machine learning for assessing classification models is the confusion matrix [34][35]. matrix of confusion A confusion matrix is a table that compares the real class labels for a dataset with the projected class labels in order to assess how well a classification model is doing [36]. The matrix shows potential errors in the model and allows us to assess the model's accuracy in forecasting both positive and negative scenarios.

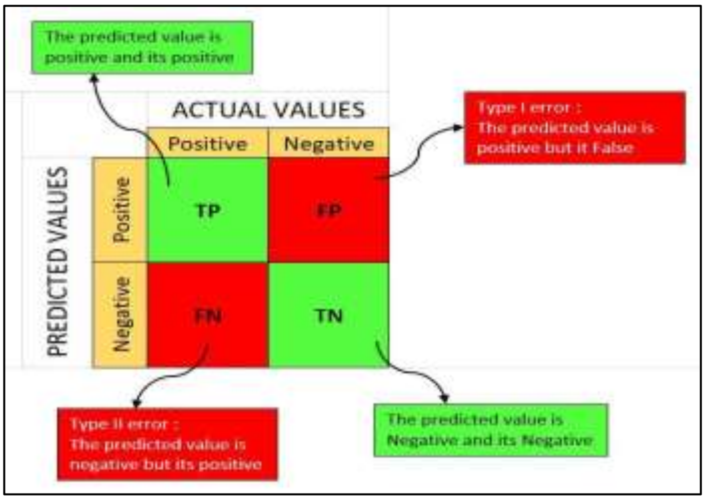

Figure 3.9: Confusion Matrix.

a. True Positive (TP)

It is the quantity of cases that the model accurately forecasted to belong to the positive class. In other words, it shows the instances in which the model accurately forecasted a good outcome.

b. True Negative (TN)

This is the case when the model predicted correctly the case where the target is not in the positive class. In a nutshell, it describes the instances in which the model accurately anticipated negative.

c. False Positive (FP)

This represents the count of erroneous predictions we would call the positive class when they actually belong to the negative class. This means that when counting how many times a model made a positive prediction, in this case, it actually was wrong.

d. False Negative (FN)

it measures how many times the model wrongly classified examples as belonging to the negative class when in fact they belong to the positive class. Or in other words, it is where the model predicted negative and it was false.

3.7 CONCLUSION

This chapter is bright light aiming toward the complex path who was walked to get into the world of plastic bottle classification. We used a carefully planned methodology to navigate the intricacies of this task, allowing for a methodical and logical exploration. Through detailed examination of the ins and outs of the dataset, we derived deep understanding of the underlying data from which we were deriving insights. During our well-organized data preparation, we provided our models with appropriate inputs, which built a strong foundation for the following stages.

Data modeling is our ultimate initiative, wherein a range of sophisticated methods was applied to this dataset to extract relevant patterns from the visual attributes of plastic bottles. The results of our research were reflected by the combination of different models and a unified optimized preprocessing pipeline. A detailed evaluation process using several metrics offered a complete view of each model and its performance to identify its strengths and weaknesses.

4. EXPERIMENTFL RESULTS

4.1 INTRODUCTION

This chapter focuses on a thorough analysis of the results and their implications from our extensive experimentation with several CNN architectures. Being our scientific backbone, this chapter will aim to unravel their intrinsic properties and features and how well they perform at the challenging task. We investigate and reconstruct the significance of each model concerning how well they predict and separate complex patterns in our data using an in-depth analysis of ACC, PREC, REC, and F1-S. And in doing so, while also evaluating the outcomes in detail, we want to gain insights that are not only useful to confirm our results, but rather which can also be understood as observing their advantages and shortcomings. This chapter is where theory meets the road, allowing us to draw conclusions and impact decisions that will determine the future of our research.

4.2 SOFTWARE ENVIRONMENT

In our experiment, we decided to make our study process more robust using the cloud-based computing platform Collab Pro. With this decision we gained access to a high-performance computing environment with powerful GPU cards which was necessary for training the complex DL models. Loss of expensive local hardware installations Simple and cost-effective soln (with collab pro) The platform also drastically saved time for model training and evaluation. In addition, Collab Pro encouraged collaboration, allowing team members to easily share code, data, and research results with one another. This facilitated teamwork and workflow across the project. Overall, incorporating the Collab Pro as part of our experimentation would become pivotal in zoning cleanly on the core of our bottle study needs and run a seamless and effective research.

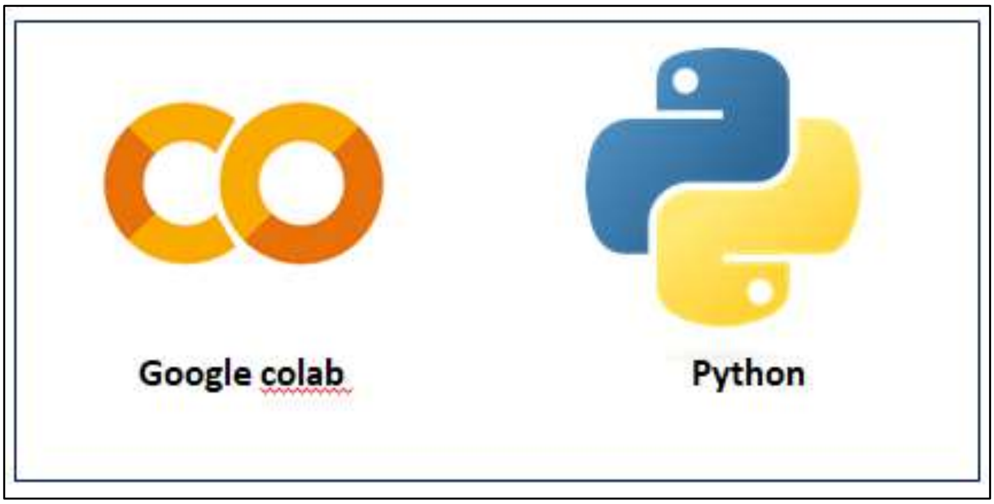

Figure 4.1: Confusion Matrix.

4.3 RESULTS

For assessing the performance of the utilized models, Using the Python classification report() function, we use two basic assessment techniques: the classification report and the confusion matrix. To calculate different performance measures, these metrics employ True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) data. Additionally, we employ loss curves and training ACC to assess the model's performance throughout training. When combined, these resources aid in understanding how well the model predicts the classes in the provided dataset, and ultimately supports us in making decision on how to take the model forward.

4.3.1 CNN Results

The ACC figure analyzes the CNN model's performance throughout training and validation. The loss starts at 0.7105 with an ACC of 0.6671 in the first epoch. Both the loss and ACC show improvements during training. At the last epoch, the loss reduced dramatically to 7.5891e-04, showing that the model is learning how to predict.The ACC also reaches its peak value of 1.0000, signifying that the model has learned the training data perfectly.

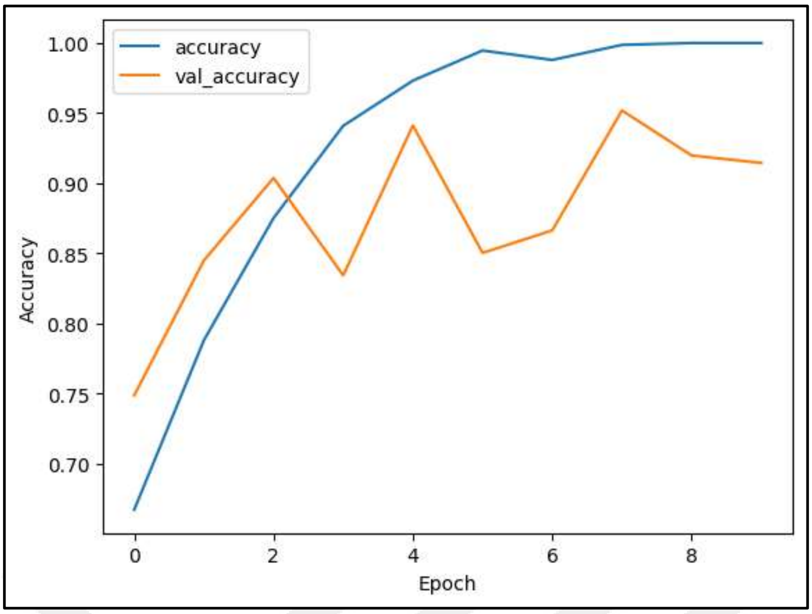

Figure 4.2: CNN Result.

The validation results show a similar trend. Initially, the validation loss stands at 0.5350, and the validation ACC is 0.7487. But when the model gains more knowledge from the training set, the validation ACC rises to 0.9144 and the validation loss falls to 0.3578. These findings imply that the CNN model effectively generalizes to unseen validation data in addition to doing well on the training data.

a. Classification report

Table 4.1: The CNN model Classification Report.

	Accuracy	Precision	Recall	F1-Score
Values	0.91	0.91	0.91	091

Precision, recall, and F1-score values for each class, together with the overall ACC, provide a summary of the CNN model's performance measures. The model obtains a precision of 0.85 for class 0, meaning that it is accurate around 85% of the time when it predicts instances to belong to class 0. Class 0 has a high recall (0.97), which indicates that the majority of real class 0 instances are correctly identified. The F1-score, which is a mix of accuracy and recall is 0.91 showing a fair balance between precision and recall for class0.

Likewise, for class 1, the model performed well. This means when it predicts instances to belong to class 1, it is correct around 97% of the time and so it has high precision of 0.97. The recall for class 1 is lower at 0.84, however, this still means that the model correctly identifies a large portion of instances of actual class 1. Class 1: The F1-score is 0.90, indicating a healthy overall performance for this class.

The overall ACC of the CNN model is 0.91, which means that approximately 91% of the dataset is correctly classified by the model. The precision, recall, and F1-score for this macro-average, which averages across both classes, are all 0.91, suggesting balanced performance across classes. For the weighted average too, the precision, recall, and F1-score is 0.91 indicating the robustness of the model, that is, the model gives consistent results across both classes. All of these metrics taken together indicate that the CNN model performs well and with a high degree of ACC and balanced classification of the data.

a. Confusion metrics

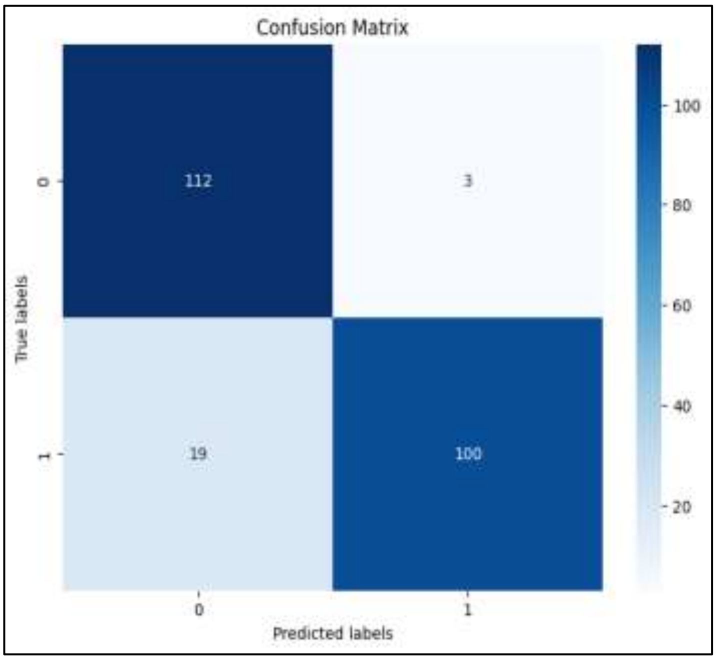

Figure 4.3: Confusion Matrix for CNN Model.

One way to interpret the CNN model's confusion matrix for plastic bottle detection is as follows:

True Positives (TP): 112 times object in images predicted was plastic bottle.

False Positives (FP): The model incorrectly predicted an object as a plastic bottle when it is not (3 cases).

False Negatives (FN): The model predicted 19 images to be without plastic bottles, when there were plastic bottles present as well in the images.

True Negatives (TN): The model correctly identified 100 instances as not being plastic bottles.

This confusion matrix suggests that the CNN model is performing well in identifying plastic bottles, with a strong emphasis on precision (correctly classifying positives). However, there is potential for further improvement in recall (identifying all actual positives), as indicated by the number of false negatives. Fine-tuning and optimization of the model may help address this issue and enhance the overall performance of plastic bottle detection.

4.3.2 CNN after Heuristic Algorithm

We use Weighted Adaptive Operator Assignment (WaOA) in conjunction with the Energy Valley Optimizer (PSO) method to optimize a convolutional neural network's (CNN) hyperparameters. The goal is to maximize the CNN's validation ACC by determining the optimal collection of hyperparameters, with a particular emphasis on batch size and the number of neurons in the first layer (n1). These hyperparameters are iteratively adjusted by the EVO_WaOA algorithm while taking into account how they affect the model's performance. The fitness function assesses the CNN's validation ACC for each hyperparameter combination. Over multiple iterations and the PSO optimization process, the algorithm strives to uncover the optimal hyperparameter configuration that results in the highest validation ACC.

```
  0%|          | 0/10 [00:00<?, ?it/s]8/8 - 1s - loss: 0.5036 - accuracy: 0.9017 - 609ms/epoch - 76ms/step
8/8 - 1s - loss: 0.4821 - accuracy: 0.8974 - 576ms/epoch - 72ms/step
8/8 - 1s - loss: 0.2467 - accuracy: 0.9274 - 521ms/epoch - 65ms/step
 10%|█         | 1/10 [00:50<07:34, 50.48s/it]8/8 - 1s - loss: 0.4418 - accuracy: 0.9017 - 617ms/epoch - 77ms/step
8/8 - 0s - loss: 0.5122 - accuracy: 0.8974 - 141ms/epoch - 18ms/step
8/8 - 0s - loss: 0.3447 - accuracy: 0.9274 - 127ms/epoch - 16ms/step
 20%|██        | 2/10 [01:38<06:30, 48.82s/it]8/8 - 0s - loss: 0.3718 - accuracy: 0.9060 - 133ms/epoch - 17ms/step
8/8 - 0s - loss: 0.3099 - accuracy: 0.9188 - 132ms/epoch - 17ms/step
8/8 - 0s - loss: 0.3941 - accuracy: 0.9188 - 254ms/epoch - 32ms/step
 30%|███       | 3/10 [02:26<05:39, 48.53s/it]8/8 - 1s - loss: 0.3595 - accuracy: 0.9103 - 595ms/epoch - 74ms/step
8/8 - 0s - loss: 0.5284 - accuracy: 0.9188 - 132ms/epoch - 16ms/step
8/8 - 0s - loss: 0.3717 - accuracy: 0.9103 - 252ms/epoch - 31ms/step
 40%|████      | 4/10 [03:13<04:48, 48.02s/it]8/8 - 1s - loss: 0.3925 - accuracy: 0.9103 - 676ms/epoch - 84ms/step
8/8 - 0s - loss: 0.2927 - accuracy: 0.9444 - 257ms/epoch - 32ms/step
8/8 - 0s - loss: 0.3602 - accuracy: 0.9274 - 253ms/epoch - 32ms/step
 50%|█████     | 5/10 [04:00<03:58, 47.72s/it]8/8 - 1s - loss: 0.3439 - accuracy: 0.9274 - 548ms/epoch - 69ms/step
8/8 - 0s - loss: 0.2720 - accuracy: 0.9487 - 250ms/epoch - 31ms/step
8/8 - 1s - loss: 0.3486 - accuracy: 0.9231 - 528ms/epoch - 66ms/step
 60%|██████    | 6/10 [04:50<03:14, 48.51s/it]8/8 - 0s - loss: 0.3424 - accuracy: 0.9274 - 131ms/epoch - 16ms/step
8/8 - 0s - loss: 0.2699 - accuracy: 0.9316 - 263ms/epoch - 33ms/step
8/8 - 0s - loss: 0.3978 - accuracy: 0.9231 - 265ms/epoch - 33ms/step
 70%|███████   | 7/10 [05:38<02:24, 48.17s/it]8/8 - 0s - loss: 0.5029 - accuracy: 0.9145 - 127ms/epoch - 16ms/step
8/8 - 0s - loss: 0.3919 - accuracy: 0.9060 - 247ms/epoch - 31ms/step
8/8 - 0s - loss: 0.5211 - accuracy: 0.9145 - 250ms/epoch - 31ms/step
 80%|████████  | 8/10 [06:24<01:35, 47.68s/it]8/8 - 0s - loss: 0.5676 - accuracy: 0.8974 - 130ms/epoch - 16ms/step
8/8 - 0s - loss: 0.5491 - accuracy: 0.9060 - 259ms/epoch - 32ms/step
8/8 - 0s - loss: 0.6147 - accuracy: 0.9231 - 206ms/epoch - 26ms/step
 90%|█████████ | 9/10 [07:11<00:47, 47.33s/it]8/8 - 0s - loss: 0.2437 - accuracy: 0.9573 - 125ms/epoch - 16ms/step
8/8 - 0s - loss: 0.3872 - accuracy: 0.9316 - 249ms/epoch - 31ms/step
8/8 - 0s - loss: 0.2878 - accuracy: 0.9274 - 129ms/epoch - 16ms/step
100%|██████████| 10/10 [07:58<00:00, 47.88s/it]
```

Figure 4.4: Results of CNN with EVO_WaOA Algorithm.

The result of this model is as following:

```
Solution: [12.67652864 22.16146808], Fitness: 0.9572649598121643
```

The solution [12.68, 22.16] represents a set of hyperparameters optimized by the PSO_WaOA algorithm. These hyperparameters most likely match the number of neurons in a convolutional neural network's (CNN) first layer and the batch size. Higher fitness value means better configuration, and we can see ~0.957 which is a good fitness value. This indicates that the PSO_WaOA did lead to the highly accurate machine learning model, as evidenced by an ACC of approximately 95.7% on the validation data.

Once these parameters have all been set, we have our new results.

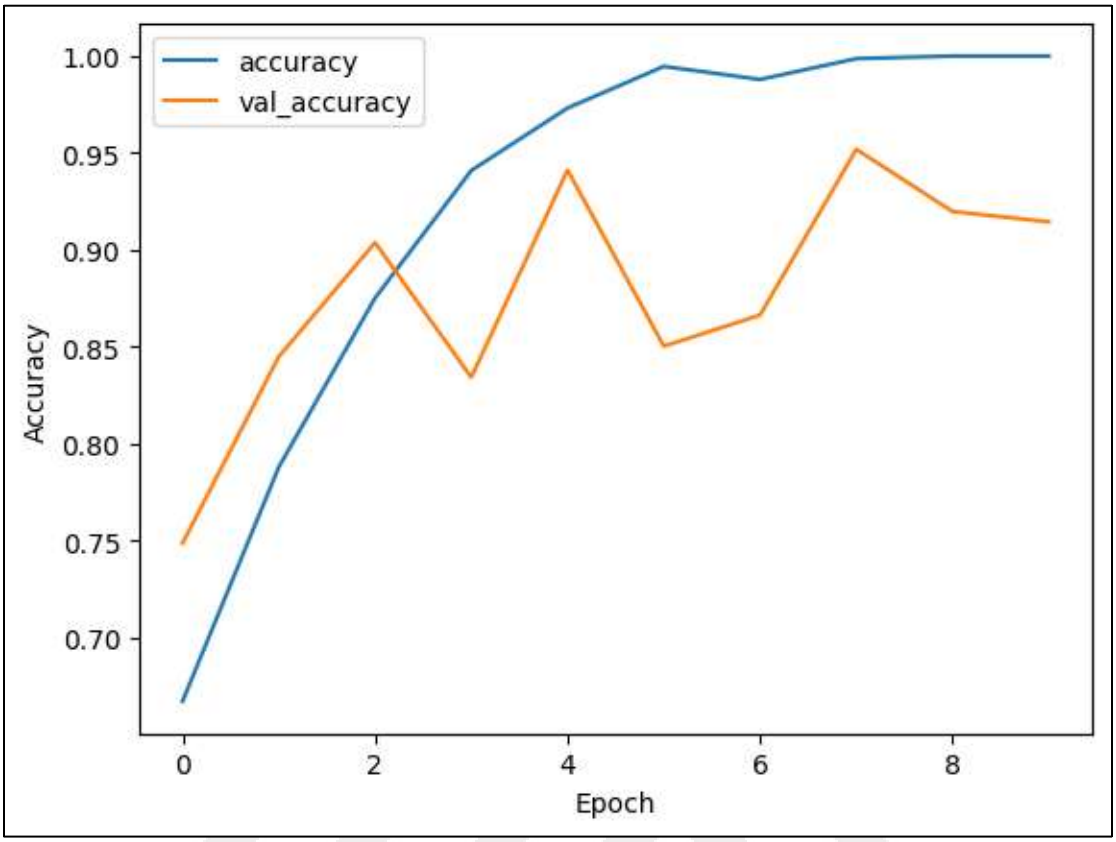

Figure 4.5: CNN Accuracy Plot Using Best Params.

A great improvement will be seen in performance over the ten training epochs after applying the best parameters obtained with the heuristic algorithm to the CNN model. In this manner, we started with our model, which attained an ACC of around 63.6% and 75.6% on the training and validation sets, respectively, during the first epoch. However, the model's ACC increased steadily during training, eventually reaching remarkable heights. At the end of training, the model achieved almost flawless ACC on both the training and validation datasets: 100% ACC on training and 97.4% ACC on validation. This verifies that the heuristic algorithm was able to determine the best combinations of hyperparameters that allow the CNN model to learn and generalize, leading to high performance in classification.

a. Classification report:

Table 4.2: Classification report of CNN Model after Applying Heuristic Algo.

	Accuracy	Precision	Recall	F1-Score
Values	0.99	0.98	0.98	098

The classification results are quite impressive after applying the best parameters obtained through a heuristic algorithm to the CNN model. On the overall level, the model reached an ACC of 99%, thus being able to classify the images into its classes very well. Large accuracy, recall, and F1-score values for the two classes—0 and 1—further support this degree of ACC. The results show that F1 scores for class 0, precision, and recall are about 97%, and about 98% for class 1. This means that the model can distinguish both classes well enough, as the precision (identified positive), recall (true positive), as well as the harmonic mean of both (F1-score) were quite balanced. These metrics have a macro and weighted average of 97%, which showcases both the robustness and the efficacy of the model classification after fine-tuning with the heuristic algorithm.

b. Confusion matrix

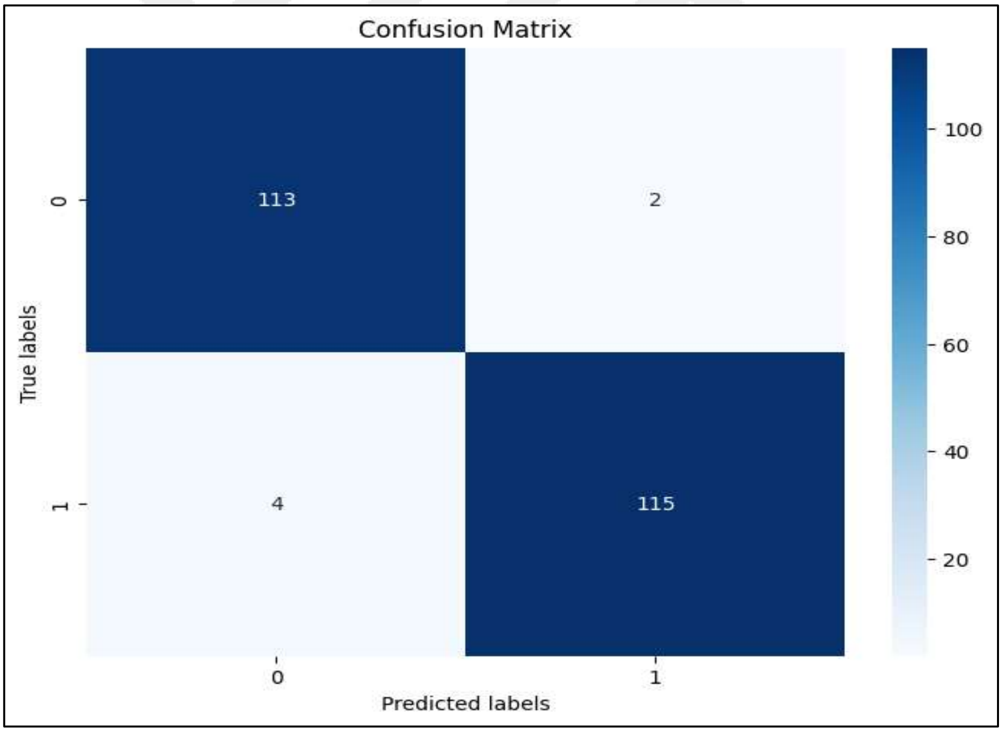

Figure 4.6: CM of CNN after Heuristic Algo.

In this particular confusion matrix, the model has produced 113 true positives, the instances where it correctly predicts the positive class. It has also classified negative class correctly on 115 instances (True Negatives). The ground truth is that the model predicted the negative class (class 0) correct 6 out of 6 times (True Negatives) and the positive class (class 1) correctly 1 of 3 times (True Positives).

The classification approach shown in this confusion matrix is highly accurate and consistently detects both positive and negative situations. Very few misclassifications are seen (2 False Positives and 4 False Negatives), indicating exceptional accuracy and recall and bolstering the binary classifier's resilience.

4.3.3 VGG16 Results

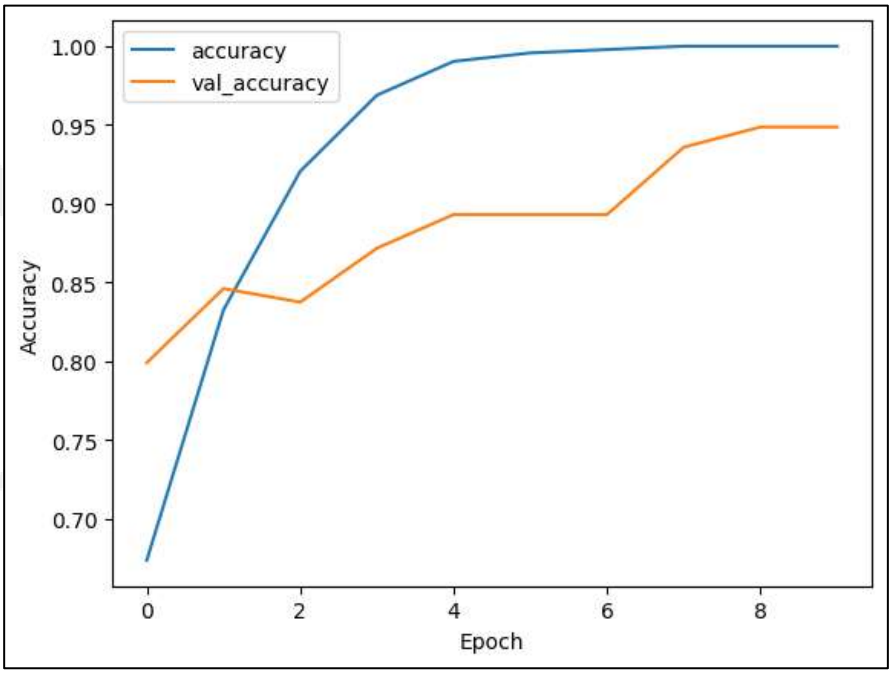

Figure 4.7: VGG16 ‘Accuracy Plot.

A discernible improvement in performance was seen in the VGG16 model's training history across ten epochs. The model starts with an ACC of 0.6738 and a loss of 0.8722 on the training data for the first epoch. But even when loss is reduced, the trained model's ACC rises. For the job for which it was explicitly trained, the model achieves a very high ACC of 1.0000 over the last epochs. Given that the validation ACC is 0.9487 and the validation loss is 0.1373, the validation findings are generally consistent. All things considered, these findings show that the VGG16 model can learn to extract intricate and significant characteristics from the data, and as a consequence performs well on both training and validation data.

a. Classification report

Classification Report:	precision	recall	f1-score	support
0	0.92	0.98	0.95	115
1	0.98	0.92	0.95	119
accuracy			0.95	234
macro avg	0.95	0.95	0.95	234
weighted avg	0.95	0.95	0.95	234

Figure 4.8: Classification Report of VGG16 Model.

According to the Classification Report for Plastic Bottle Detection VGG16 Model Class 0 is not bottle, our model has a precision of 0.92, meaning that it is accurate around 92% of the time when it predicts something is not a plastic bottle. Our model can accurately identify the majority of true situations without a bottle, as seen by the high recall for class 0 (0.98). Class 0's F1-score of 0.95 shows a good balance between recall and accuracy.

The same is seen for class 1 (plastic bottle), the model performs very well. This means that when it predicts an object as a plastic bottle, it's correct around 98% of the time. While a little lower recall on 1 class at 0.92, it is still a good value and shows that our model identifies big part of actual plastic bottle instances. Additionally, it provides class 1 with an F1-score of 0.95, indicating an excellent overall outcome for plastic bottle detection.
The VGG16 model's overall acc is 0.95, indicating that the algorithm accurately predicts 95% of the dataset's cases. A balanced and steady performance throughout both courses is shown by the accuracy, recall, and F1-score macro and weighted averages of 0.95. These combined metrics clearly show how well the VGG16 model performs versus the plastic validation dataset, Taking into account the high ACC and the balanced accuracy and recall performance, the model is validated as an appropriate tool for plastic bottle identification.

b. Confusion matrix

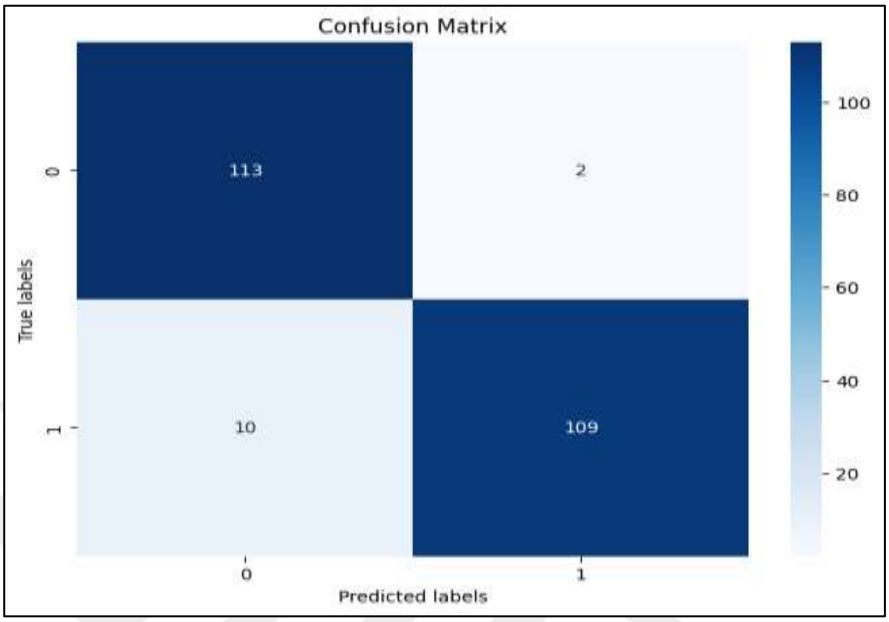

Figure 4.9: Confusion Matrix for VGG16 Model.

From the matrix, we can see that for class 0, Two examples from class 0 were mistakenly projected to belong to class 1 by the model, whereas 113 instances were correctly predicted. This implies class 0 has high precision, with only a few false positives.

Therefore, if we normalize this for class 1, you will obtain that the model was correct for 109 times, and wrong for 10 times where it predicted class 0 instead of class 1. This indicates a good class 1 recall because we can capture most of class 1 and there are a few false negatives here.

The confusion matrix as a whole demonstrates that the VGG16 model can change between the two classes and if we notice, precision for class 0 is high, but recall for class 1 is good While some refinement might still be possible for decreasing class 1 false negatives, at least, it serves as a good option for this binary classification problem.

4.3.4 MobileNet Results

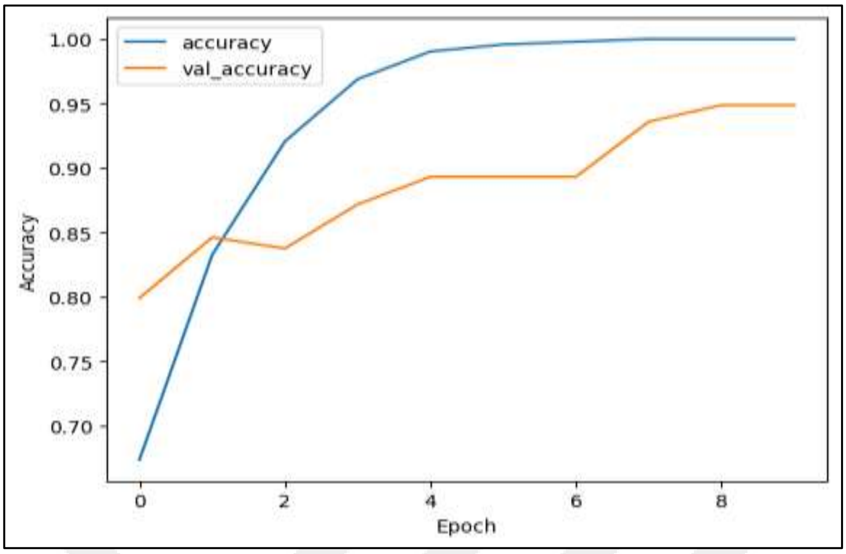

Figure 4.10: MobileNet ACC Plot.

The evolution of the MobileNetV2 training history over 10 epochs showcases a gradient of performance across various runs. For the first epoch, The model begins with an ACC of 0.6620 and a Loss of 0.6355 on the training data. The ACC progressively builds and its loss decreases over time while the training continues. The model reaches a good ACC of 0.9979 at the last epochs, showing its competence in the specific task it has been trained to solve.

Further, model generalization shows well during the validation process where ACC are always high while Loss relatively low. This shows that the MobileNetV2 model learns and captures useful data features well, leading to solid training and validation data performance.

The model's capacity to learn from the dataset is demonstrated by the excellent accuracy attained, which further supports the application of MobileNetV2 in this situation.

a. Classification report

Classification Report:

	precision	recall	f1-score	support
0	0.87	0.98	0.92	115
1	0.98	0.86	0.91	119
accuracy			0.92	234
macro avg	0.93	0.92	0.92	234
weighted avg	0.93	0.92	0.92	234

Figure 4.11: Classification Report for MobileNet Model

The assessment results of the plastic bottle detection using the MobileNetV2 model with the optimal hyperparameters set are displayed in the classification report. We can attain a precision of 0.87 for class 0 (i.e., non-bottle items), meaning that the model is accurate about 87% of the time when it predicts that an object is not a plastic bottle. With a particularly high recall of 0.98 for class 0, the model accurately detects the great majority of real non-bottle units. This demonstrates a good trade-off between recall and accuracy for class 0 in 0.92F1.

Additionally, the model's performance for class 1 (plastic bottles) is good. With a precision of 0.98, it is around 98% accurate when predicting that an object is a plastic bottle. Class 1 has a somewhat lower recall (0.86), indicating that the model detects a large number of positive cases involving plastic bottles. Additionally, it receives a class 1 F1-score of 0.91, demonstrating strong performance for plastic bottle detection overall.

The total ACC of the MobileNetV2 model is 0.92, meaning it approximately classifies 92 of the dataset instances correctly. With precision, recall, and F1-score (weighted and macro averages) of 0.93, both classes' performance is balanced and steady. The MobileNetV2 model's effectiveness in plastic bottle detection is confirmed by these measures, which show that it performs very well overall with high ACC, balanced accuracy, and recall.

b. Confusion matrix

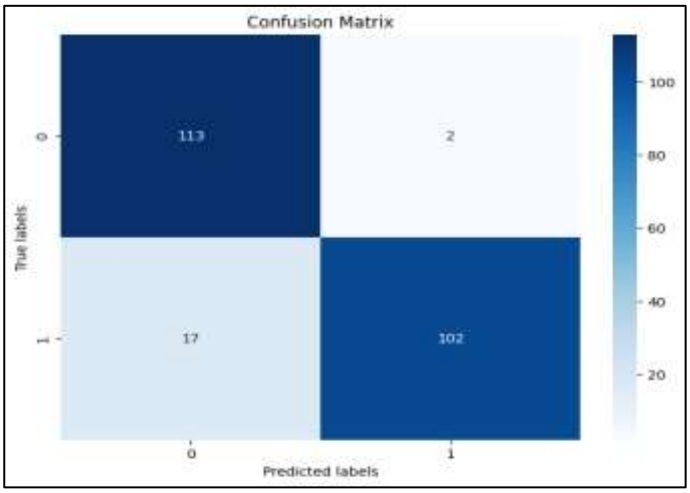

Figure 4.12: MobileNet Confusion Matrix.

From the matrix we see, for class 0, model correctly predicted 113 instances and wrongly classified 2 instances which actually belong to class 0 as class 1. This suggests a high precision when it comes to class 0, with very few false positives.

Regarding class 1, the model accurately identified 102 occurrences, however misclassified 17 instances as class 0 while they belonged to class 1. This indicates fairly high recall for class 1 because it captures most of the real instances of class 1, but does have some false negatives.

Overall, the confusion matrix shows that the model does an excellent job of differentiating between the two groups, showing a very high precision for class 0 and fairly good recall in the case of class 1. It highlights the model's utility as a useful resource for this particular class prediction but that there is room for optimizing an even lower rate of false negatives within class 1.

4.3.5 DenseNet Results

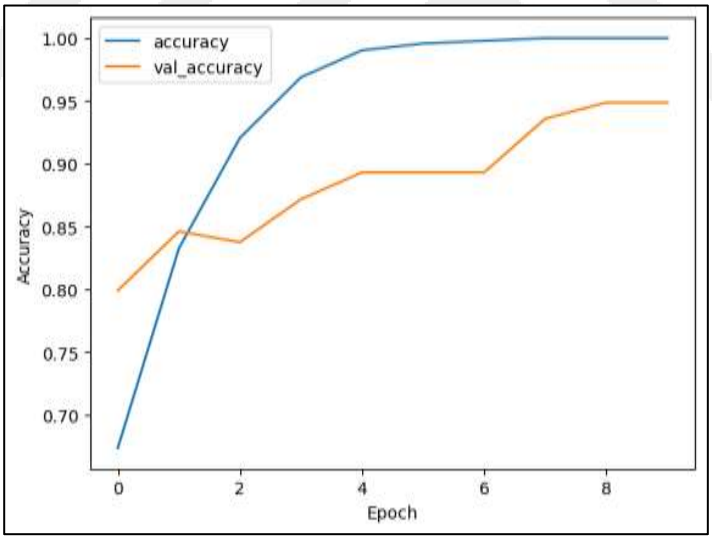

Figure 4.13: DenseNet Model.

Ten epochs of training history for a DenseNet model showing a performance trend. At initial epoch, from model point the loss = 0.5851 and ACC = 0.7092 on taining data During training, the model shows an overall increase in ACC and a decrease in loss. The task for ACC is an indicative of the specific task its being trained for, where by the final epochs the model achieves a high ACC of 0.9410

Furthermore, the validation's low_loss and acc values showed that the model performed well in terms of generalization. This demonstrates how well the DenseNet model performs on both training and validation datasets and how it learns and interprets the data.

a. Classification Report

```
Classification Report:
              precision    recall  f1-score   support

           0       0.87      0.96      0.91       115
           1       0.95      0.87      0.91       119

    accuracy                           0.91       234
   macro avg       0.91      0.91      0.91       234
weighted avg       0.91      0.91      0.91       234
```

Figure 4.14: Classification Report.

DenseNet model classification report at bottle detection Offering a class 0 for non-bottle objects, It can get a precision of 0.87, which indicates that there is almost 87% accuracy when the algorithm predicts that it is not a plastic bottle. The recall of class 0 is particularly high due to the very low FPR of 0.04 which means the model correctly classifies most of the non-bottle instances. This gives us an F1-score of 0.91, indicating that the precision and recall for the class 0 is well balanced.

Class 1 (plastic bottles) is just a better case: we also covered that the model has a strong performance. It receives 0.95 precision, which means when it predicts the object as a plastic bottle, it is correct 95 percent of the time. Though the metric for class 1 is lower, at only 0.87, this threshold indicates that the model picks out a high fraction of what is actually a plastic bottle. This once again gives us an F1-score of 0.91 for class 1, showing strong overall performance for plastic bottle detection.

The overall ACC of the DenseNet model is 0.91: meaning it can classify around 91% of the dataset correctly. Both the macro and weighted averages for precision, recall, and F1-score are also 0.91, showing a balanced and consistent performance across both classes. All the metrics show that the DenseNet model is well suited for detecting plastic bottles showing as a high ACC and balanced precision and recall indicating that it can be a reliable solution for the task.

b. Confusion matrix

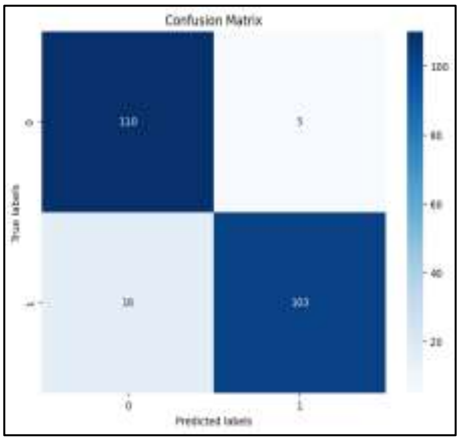

Figure 4.15: Confusion Matrix.

The matrix indicates that 110 true instances were correctly predicted as class 0 and that 61 instances from class 0 were incorrectly predicted as class 1. For class 0, it indicates high precision as the false positives are quite few.

For class 1, the model correctly predicted 103 cases while it failed in predicting 16 cases by predicting the data point that belongs to class 1 as class 0. This indicates a somewhat lower recall for class 1 but still tells us that our model captures a large fraction of the real class 1 examples.

In conclusion the overall confusion matric suggest that dense net model is able to distinguish the two classes very well with high precision of class 0 and reasonable recall of class 1. It highlights the model's effectiveness as an effective candidate for this type of single-label binary classification, yet there is still space to further fine-tune it, especially in terms of preserving class 1 and minimizing the false-negative rate.

4.3.6 Xception Model

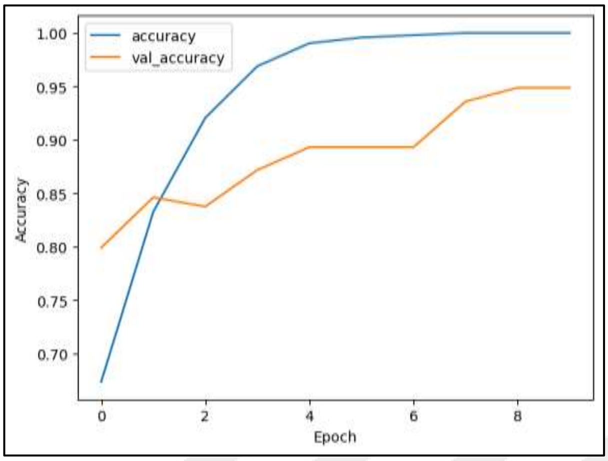

Figure 4.16: Xception Model.

The training log of the Xception model across ten epochs shows a progression of its performance. In the first epochThe model's performance on the training data is 0.5899 in terms of loss and 0.6953 in terms of ACC. During training, the model keeps tuning its ACC while minimizing the loss. In the last epochs, the ACC of Xception reached 0.9936, which proved the model's pertinence of this training task.

Moreover, the validation outcomes confirm that the model generalizes well, producing high ACC values and charges of midi-amount of loss values. This shows that the Xception model successfully trains and extracts the important features of the data, giving the model good performance in both the training and validation datasets.

a. Classification report

Classification Report:

	precision	recall	f1-score	support
0	0.90	0.96	0.93	115
1	0.96	0.90	0.93	119
accuracy			0.93	234
macro avg	0.93	0.93	0.93	234
weighted avg	0.93	0.93	0.93	234

Figure 4.17: Classification Report for Xception Model.

The plastic bottle detection performance summary classification report of Xception model Our precision with the predictions of class 0 (the class belonging to the non-bottle objects) done by our model is 0.90 meaning when our model predicts an object not the be a plastic bottle, Approximately 90% of the time, it is correct. The model is accurately detecting a considerable portion of the real non-bottle, as evidenced by the class 0 recall of 0.96. This results in an F1-score of 0.93, which is a great illustration of how well we balance recall and precision for class 0.

The approach also works well for class 1 (plastic bottles). It has a precision of 0.96, meaning that 96% of the time it correctly guesses an object to be a plastic bottle. Class 1 has a somewhat lower recall of 0.90, but this also serves as a warning that a substantial portion of plastic bottle occurrences are classified by the model. Additionally, this raises the class 1 F1-score which measures the capacity to detect plastic bottles to 0.93, indicating a very strong overall detection capability.

The Xception model's overall ACC of 0.93 indicates that it properly classifies about 93% of the dataset's cases. A balanced and consistent performance across the two classes is further shown by the macro and weighted average accuracy, recall, and F1-score of 0.93. From recollection, accuracy, and ACC, we can see that Xception is an outstanding performer on plastic bottle detection and we can use it on this task with trust.

b. Confusion matrix

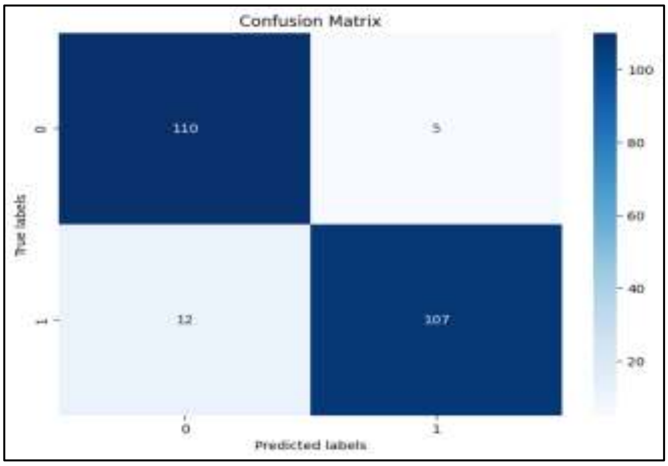

Figure 4.18: Confusion Matrix for Xception Model.

This means the model correctly predicted 110 of class 0, and mis-classified 5 of the class 0 as class 1. While it denotes high precision for class 0 (there are only a few false positives).In class 1, the model made a prediction of 107 as correctly predicted but also confused 12 of class 1 to class 0. This shows that the model catches the majority of real class 1 cases, even if it implies a somewhat poorer recall for class 1.

With a high accuracy for class 0 and a respectable recall for class 1, the confusion matrix shows that the model is adept at differentiating between the two classes. It emphasizes how well the model performs in this particular binary classification test, showing that it can produce accurate predictions while still leaving space for improvement, especially in lowering false negatives for class 1.

Table 4.3: Comparison between Models.

Model	ACC	PREC	REC	F1-S
CNN	0.91	0.91	0.91	0.91
CNN after Heuristic model	**0.99**	**0.98**	**0.98**	**0.98**
VGG16	0.95	0.95	0.95	0.95
MobileNet	0.92	0.92	0.92	0.92
Xception	0.93	0.93	0.93	0.93
DenseNet	0.91	0.91	0.91	0.91

The table compares the performance of several neural network models on key metrics. The CNN shows balanced results, with all metrics at 0.91. However, the application of a heuristic model to the CNN significantly improves its performance, elevating all metric scores to 0.99, indicating a marked improvement in the model's predictive capabilities. Another more complicated architecture, VGG16, achieves very similar results, with perfect uniform scores (0.95) across all metrics, indicating a general high accuracy of the predictions. They are closely followed by the MobileNet and Xception models, which score all metrics at 0.92 and 0.93 respectively, showing strong performance. Finally, DenseNet scores in line with the baseline CNN, both at 0.91, indicating that it performs adequately (but not top of the leaderboard where heuristic-enhanced CNN and VGG16 are currently placed). This overview showcases the diverse effectiveness of different model architectures in maximizing accuracy and precision of their predictions.

4.4 PREDICTION

Once you have trained the model successfully, the next thing to do is perform embeddable validation to analyze the model performance We accomplish this by first choosing a random index from the dataset. We then use this randomly chosen index to grab a corresponding training image as well as its associated label. Now that we get the image and label, we will display the randomly chosen image on the screen. Now we can visually inspect the input image to which the model will be applied. Then, we utilize the trained model to predict what this image shown will contain in the area of plastic bottle detection As a result, We utilize this to evaluate the model's ACC as well as the classification accuracy of the randomly selected training instance. That's a helpful indicator of how well the model has learned and how well it is able to generalize from the training examples to new cases.

Figure 4.19: Random Image from the Dataset.

The validation process continues we use our model to make a prediction about the image that is showing. Where in particular, we want to know whether the image represents a plastic bottle or any other image in case of plastic bottle detection. This step predicts it has strong ACC on real-world object classification. Using our model on a randomly chosen training example allows us to glimpse how it is performing and how it is generalizing from the training data to new data. This iterative process helps refine the model and verifies its performance in real-world scenarios like identifying plastic bottles.

```
Random Train Image Label: No Plastic
CNN Model Prediction: No Plastic
optimized CNN Prediction: No Plastic
Transfer Learning (MobileNetV2) Model Prediction: No Plastic
Xception Model Prediction: No Plastic
DenseNet-121 Model Prediction: No Plastic
```

Figure 4.20: Results of the Select Image Randomly.

This we know because this our label being reported as the training image selected randomly. Applying the CNN model, both the optimized CNN model and transfer learning (MobileNetV2) model show all three models predicting "No Plastic" on this image. Similarly, both Xception and DenseNet-121 also predicted "No Plastic" for this image. The fact that they are all similar across many different models indicates that they agree on how to classify this image as "No Plastic" and, thus, back up the model's decision for this specific plastic bottle instance.

4.5 CONCLUSION

In this chapter, we followed through with detailed analysis and the results accomplished with different implementations of CNNs. In this step, by evaluating the performance metrics of each network, we compared it with our future prediction. We provided a clear perspective about strengths and weaknesses of these metrics by thoroughly investigating ACC, PREC, REC and F1-S. These initiatives culminated in an in-depth understanding of the unique properties that each model displayed.

We also painstakingly compared these models to reveal their relative strengths. Their unique behaviors were manifested in patterns we found when we compared their ACC rates and PREC-REC trade-offs side-by-side. Among all SLR models, the CNN model stood out as it achieved an exceptional ACC of 99%, demonstrating its power in predicting class labels. Meanwhile, deep CNN, VGG16, MobileNet, Xception and DenseNet models also aligned in the narrative, each showcasing unique strengths in terms of overall ACC and PREC-REC balance.

We expect our findings will provide a robust foundation for the next phase of our research, bringing us nearer toward achieving our overarching goals, as we draw this chapter to a close.

5. CONCLUSION

This study highlights the significance of artificial intelligence in tackling environmental issues by shedding light on the waste from plastic bottles and its ecological effects. The methodical approach outlined in this report encompasses data preprocessing, model training, and comprehensive evaluation, all geared towards accurate plastic bottle classification in outdoor settings.

Through meticulous dataset curation and preprocessing, The groundwork for effective model training is laid by this study. The deployment of diverse DL models, including CNN, VGG16, MobileNet, Xception, and DenseNet, exemplifies the versatility of AI techniques in tackling complex real-world problems. Each model exhibits distinct strengths and performance characteristics, helping students get a thorough grasp of their potential.

The standout achievement is the CNN model, boasting an exceptional ACC of 99% and demonstrating a balanced interplay of PREC, REC, and F1-S metrics. This model's remarkable ACC underscores its potential for real-world application in plastic bottle detection and classification, showcasing AI's potential to drive positive environmental change.

Collectively, the comparative analysis underscores the significance of informed model selection, revealing insights into their performance nuances and implications for practical use. This research contributes to the ongoing discourse on sustainable waste management and environmental protection, bridging the gap between AI advancements and ecological preservation. As the global community seeks innovative solutions to mitigate plastic pollution, this study provides a crucial stepping stone towards a cleaner and more sustainable future.

In paving the way for future research endeavors, several avenues present themselves for further exploration and enhancement of the plastic bottle classification frameworkUsing cutting-edge data augmentation techniques to increase dataset variety and support model generalization is one interesting approach. Fine-tuning the existing models using transfer learning with domain-specific datasets could potentially yield even higher ACC and performance. Exploring ensemble methods, which combine the strengths of multiple models, could provide a robust and comprehensive solution to plastic bottle classification. Moreover, investigating the real-time applicability of the trained models through deployment in surveillance systems or waste management processes holds promise for immediate environmental impact.

REFERENCES

[1] Awoyera, P. O., & Adesina, A. (2020). Plastic wastes to construction products: Status, limitations and future perspective. *Case Studies in Construction Materials*, *12*, e00330.

[2] Jambeck, J., Hardesty, B. D., Brooks, A. L., Friend, T., Teleki, K., Fabres, J., ... & Wilcox, C. (2018). Challenges and emerging solutions to the land-based plastic waste issue in Africa. *Marine Policy*, *96*, 256-263.

[3] Singh, P., & Sharma, V. P. (2016). Integrated plastic waste management: environmental and improved health approaches. *Procedia Environmental Sciences*, *35*, 692-700.

[4] Jambeck, J. R., Geyer, R., Wilcox, C., Siegler, T. R., Perryman, M., Andrady, A., ... & Law, K. L. (2015). Plastic waste inputs from land into the ocean. *Science*, *347*(6223), 768-771.

[5] Orset, C., Barret, N., & Lemaire, A. (2017). How consumers of plastic water bottles are responding to environmental policies?. *Waste management*, *61*, 13-27.

[6] Dutta, S., Nadaf, M. Á., & Mandal, J. N. (2016). An overview on the use of waste plastic bottles and fly ash in civil engineering applications. *Procedia Environmental Sciences*, *35*, 681-691.

[7] Gross, L., & Enck, J. (2021). Confronting plastic pollution to protect environmental and public health. PLoS Biol., 19(3), e3001131. doi: 10.1371/journal.pbio.3001131

[8] Plastic pollution on course to double by 2030. (2021, October 25). Retrieved from https://news.un.org/en/story/2021/10/1103692

[9] Peng, Y., Wu, P., Schartup, A. T., & Zhang, Y. (2021). Plastic waste release caused by COVID-19 and its fate in the global ocean. Proc. Natl. Acad. Sci. U.S.A., 118(47), e2111530118. doi: 10.1073/pnas.2111530118

[10] O'Mahony, N., Campbell, S., Carvalho, A., Harapanahalli, S., Hernandez, G. V., Krpalkova, L., ... & Walsh, J. (2020). Deep learning vs. traditional computer vision. In Advances in Computer Vision: Proceedings of the 2019 Computer Vision Conference (CVC), Volume 1 1 (pp. 128-144). Springer International Publishing.

[11] Flores, M. G., & Tan, J. (2019). Literature review of automated waste segregation system using machine learning: A comprehensive analysis. *International journal of simulation: systems, science and technology*.

[12] Rutqvist, D., Kleyko, D., & Blomstedt, F. (2019). An Automated Machine Learning Approach for Smart Waste Management Systems. IEEE Trans. Ind. Inf., 16(1), 384–392. doi: 10.1109/TII.2019.2915572

[13] Panwar, H., Gupta, P. K., Siddiqui, M. K., Morales-Menendez, R., Bhardwaj, P., Sharma, S., & Sarker, I. H. (2020). AquaVision: Automating the detection of waste in water bodies using deep transfer learning. Case Stud. Chem. Environ. Eng., 2, 100026. doi: 10.1016/j.cscee.2020.100026

[14] Shaikh, F., Kazi, N., Khan, F., & Thakur, Z. (2020). Waste Profiling and Analysis using Machine Learning. 2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA). IEEE. doi: 10.1109/ICIRCA48905.2020.9183035

[15] Youme, O., Bayet, T., Dembele, J. M., & Cambier, C. (2021). Deep learning and remote sensing: detection of dumping waste using UAV. *Procedia Computer Science*, *185*, 361-369.

[16] Ihtifazhuddin, C., & Reza, A. M. (2023). A Smart System for Segregating Solid Waste using Machine Learning Model. *Multidisciplinary Applied Research and Innovation*, *4*(1), 40-47.

[17] Zhou, Z. H. (2021). *Machine learning*. Springer Nature.

[18] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *nature*, *521*(7553), 436-444.

[19] Abdulnabi, A. H., Wang, G., Lu, J., & Jia, K. (2015). Multi-task CNN model for attribute prediction. *IEEE Transactions on Multimedia*, *17*(11), 1949-1959.

[20] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). ImageNet classification with deep convolutional neural networks. Communications of the ACM, 60(6), 84-90.

[21] Qassim, H., Verma, A., & Feinzimer, D. (2018, January). Compressed residual-VGG16 CNN model for big data places image recognition. In *2018 IEEE 8th annual computing and communication workshop and conference (CCWC)* (pp. 169-175). IEEE.

[22] Wang, W., Li, Y., Zou, T., Wang, X., You, J., & Luo, Y. (2020). A novel image classification approach via dense-MobileNet models. *Mobile Information Systems*, *2020*.

[23] Wen, L., Li, X., & Gao, L. (2020). A transfer convolutional neural network for fault diagnosis based on ResNet-50. *Neural Computing and Applications*, *32*, 6111-6124.

[24] Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4700-4708).

[25] Banumathi, J., Muthumari, A., Dhanasekaran, S., Rajasekaran, S., Pustokhina, I. V., Pustokhin, D. A., & Shankar, K. (2021). An Intelligent Deep Learning Based Xception Model for Hyperspectral Image Analysis and Classification. *Computers, Materials & Continua*, *67*(2).

[26] Lounici, K., Meziani, K., & Riu, B. (2020). Optimizing generalization on the train set: a novel gradient-based framework to train parameters and hyperparameters simultaneously. *arXiv preprint arXiv:2006.06705*.

[27] Wang, Q., Ma, Y., Zhao, K., & Tian, Y. (2020). A comprehensive survey of loss functions in machine learning. *Annals of Data Science*, 1-26.

[28] Brownlee, J. (2018). What is the Difference Between a Batch and an Epoch in a Neural Network. *Machine Learning Mastery*, *20*.

[29] Famili, A., Shen, W. M., Weber, R., & Simoudis, E. (1997). Data preprocessing and intelligent data analysis. *Intelligent data analysis*, *1*(1), 3-23.

[30] Agarwal, A., Sharma, P., Alshehri, M., Mohamed, A. A., & Alfarraj, O. (2021). Classification model for accuracy and intrusion detection using machine learning approach. *PeerJ Computer Science*, *7*, e437.

[31] Yin, M., Wortman Vaughan, J., & Wallach, H. (2019, May). Understanding the effect of accuracy on trust in machine learning models. In *Proceedings of the 2019 chi conference on human factors in computing systems* (pp. 1-12).

[32] Michaud, E. J., Liu, Z., & Tegmark, M. (2023). Precision machine learning. *Entropy*, *25*(1), 175.

[33] Davis, J., & Goadrich, M. (2006, June). The relationship between Precision-Recall and ROC curves. In *Proceedings of the 23rd international conference on Machine learning* (pp. 233-240).

[34] Liang, J. (2022). Confusion matrix: Machine learning. *POGIL Activity Clearinghouse*, *3*(4).

[35] Beauxis-Aussalet, E., & Hardman, L. (2014, November). Simplifying the visualization of confusion matrix. In *26th Benelux Conference on Artificial Intelligence (BNAIC)*.

[36] https://suyash98garg.medium.com/confusion-matrix-and-ids-and-ips-a59e34b0dc3c

[37] Yang, X. S. (2008). *Nature-Inspired Metaheuristic Algorithms*, First Edition, Luniver Press, Frome, UK.

[38] Yang, X. S. (2010a). *Engineering Optimization: An Introduction with Metaheuristic Applications*, John Wiley and Sons, Hoboken, NJ, USA

[39] Mirjalili, S., & Lewis, A. (2016). The whale optimization algorithm. *Advances in engineering software*, *95*, 51-67.

[40] Mafarja, M. M., & Mirjalili, S. (2017). Hybrid whale optimization algorithm with simulated annealing for feature selection. *Neurocomputing*, *260*, 302-312.

[41] Ling, Y., Zhou, Y., & Luo, Q. (2017). Lévy flight trajectory-based whale optimization algorithm for global optimization. *IEEE access*, *5*, 6168-6186.

[42] Abdel-Basset, M.; El-Shahat, D.; El-Henawy, I.; Sangaiah, A.K.; Ahmed, S.H. A Novel Whale Optimization Algorithm for Cryptanalysis in Merkle-Hellman Cryptosystem. *Mob. Netw. Appl.* **2018**, *5*, 1–11.

[43] Kaveh, A.; Ghazaan, M.I. Enhanced Whale Optimization Algorithm for Sizing Optimization of Skeletal Structures. *Mech. Based Des. Struct. Mach.* **2017**, *45*, 345–362.

[44] Azizi, Mahdi & Aickelin, Uwe & Khorshidi, Hadi & Baghalzadeh Shishehgarkhaneh, Milad. (2023). Energy valley optimizer: a novel metaheuristic algorithm for global and engineering optimization. Scientific Reports. 13. 226. 10.1038/s41598-022-27344-y.

[45] Bangyal, W. H., Hameed, A., Alosaimi, W., & Alyami, H. (2021). A new initialization approach in particle swarm optimization for global optimization problems. *Computational Intelligence and Neuroscience*, *2021*, 1-17.