PEOPLES CO-OPERATIVE ARTS AND SCIENCE COLLEGE, MUNNAD

(Affiliated To Kannur University)

December 2023

MCS3P05: CASE STUDY REPORT

# *WASTE CLASSIFIER*

*Submitted on partial fulfillment of the requirement for the award of the degree in Master of Computer Science with specialized in Artificial Intelligence from Kannur University .*

Submitted by

JISHNA.M

Reg. No: C2PSAI2005

DEPARTMENT OF MASTER OF COMPUTER SCIENCE 2022-2024

PEOPLES CO-OPERATIVE ARTS AND SCIENCE COLLEGE MUNNAD

(Affiliated To Kannur University)

CERTIFICATE

*This is to certify that this case study report entitled "Waste Classifier" is a bona fide record on partial fulfillment for the degree of the Master of Computer Science with specialized in Artificial Intelligence to the Kannur University through PG Department of Computer Science, Peoples Co-Operative Arts and Science College, Munnad, Kasaragod done by JISHNA.M (Reg.No: C2PSAI2005) in the year 2023.*

Place :

Date : Head of the department:

Examiner 1:

Examinar 2:

# Index

## Abstract

Recycling solid waste is an important step to reduce harmful impact such as sanitary and health problems resulting from the over use of landfills. Yet, recycling requires the sorting of solid waste, which is complex and expensive. In an attempt to ease this process, our work proposes a Learning approach using computer vision to automatically identify the type of waste and classify it into four main categories: Recyclable wast,Hazarddous wast,food wast,Residual wast.Our conceptual system consists of an automated recycling bin which automatically opens the lid corresponding to the type of waste identified. This work focuses mainly on the Machine Learning which can be trained for efficient identification.It is also usefull for childrens to learn.

# INTRODUCTION

Solid waste management is a growing global phenomenon which affects every single human being. Inappropriate management of solid waste leads to water contamination, disease transmission in human and animals and increase flooding due to the blocking of drains. These adverse effects have a direct consequence on economic growth and weighs heavier for developing countries. The World Bank estimates that 2 billion tonnes of solid waste have been generated worldwide in 2016, averaging to around 0.74 Kilograms of waste per person. According to the same report, it is expected that waste generation will increase by 50% in 2030 and 70% in 2050 if no proper actions are taken.

most popular waste disposal method worldwide is through landfills. This method has a very high damaging impact on the environment, with the release of toxins, leachate and greenhouse gas. Recovery through recycling and composting can considerably reduce the impact of landfills. Recycling inherently consists of a complex and expensive process to sort out various waste materials into different categories such as: metal, glass, plastic and paper. Some of the existing solutions can be classified into mechanical sorting, manual sorting, automated identification with tags and computer vision systems.

Manual sorting, mainly implemented in developing countries proves to be more efficient than mechanical sorting. Nevertheless, hand picking involves the manipulation of hazardous materials and can have various adverse health problems. The application of tags, such as barcodes and Radio Frequency Identifiers (RFID) have been used and proposed by various studies for efficient sorting of waste materials.The use of computer vision to classify a particular waste product involves the use of sensors and/or cameras, connected to a system to perform identification and an actuator to perform the sorting. We proposed  low cost and energy efficient recycling system based on bar code in . This work focuses on enhancing our solution to provide an automated recycling bin system which can automatically identify the type of waste being thrown and open the corresponding lid.

## OBJECTIVE

- Recycling is a key component of modern wast reduction and is the third component of the “reduce,Reuse,and Recycling”wast hierarchy.It promotes environmental sustainability.
- Manual sorting, mainly implemented and proves to be more efficient than mechanical sorting.
- Hand picking involves the manipulation of hazardous materials and can have various adverse health problems. The application of tags, such as barcodes and Radio Frequency Identifiers (RFID) have been used and proposed by various studies for efficient sorting of waste materials.
- The use of computer vision to classify a particular waste product involves the use of sensors and/or cameras, connected to a system to perform identification and an actuator to perform the sorting.
- It is also usefull for childrens to learn.

## Future scope

Waste classification through intelligent methods has become a key factor for human beings to achieve sustainable development. Intelligent waste classification can be applied to mobile devices, intelligent recyclable trash cans, etc.It is beneficial to the environment and improves the recycling of waste resources.The future scope of the model is to build a model detection of garbage .It will be easy to detect the garbage using image processing and then lead to a shift towards AI devices which will greate help to society.

It will be the best model to clean the city.

# SYSTEM ANALYSIS

## proposed systm

Waste is a major environmental issue due to its potential to contaminate air, water, and soil. Its generation is mainly attributed to human activities such as industrial production, construction, agricultural activities, pollution emissions, consumption, and waste disposal. The utilization of artifcial intelligence has the potential to bring about a revolution in waste management by enhancing the efectiveness of waste classifcation.

Smart bin system:

Numerous research studies on intelligent garbage bins have focused on functions automatic waste classifcation . These studies ofer a potential solution for cities to achieve an efective garbage collection system.

WASTE CLASSIFICATION :

Waste classification refers to the process of identifying and categorizing waste materials based on their characteristics and composition. This is typically done in order to properly dispose of the waste, recycle it, or find other ways to manage it in an environmentally responsible manner.

WASTE DATASETS:

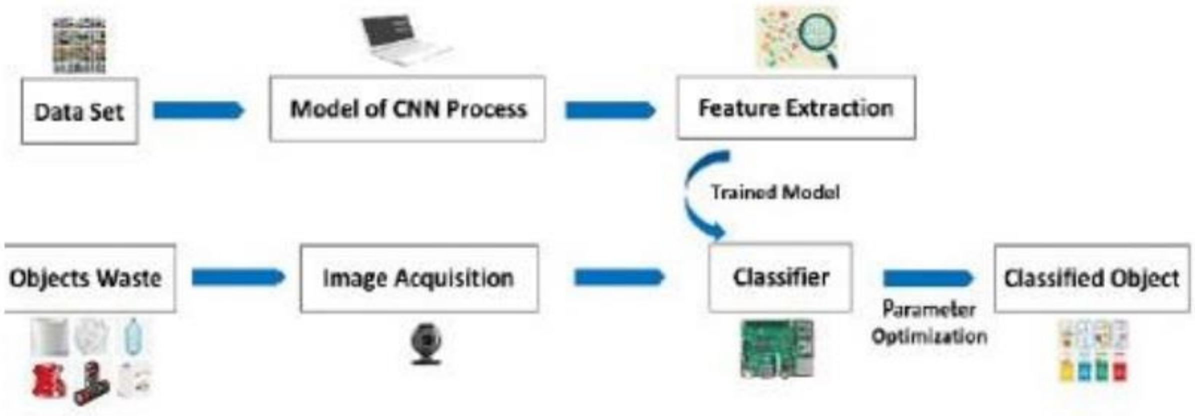
Many different methods for classifying wastes into different categories. The use of Machine Learning and Deep Learning Algorithms are the most popular approaches for it. Different wasts bins are:

Residual Waste
Food Waste
Hazardous Waste
Recyclable Waste

Differtnt wast images are:

Battery
Vegetable leaf
Apple
Zip-top cans
Newspaper
Old shoes
Watercolor pen
Disinfectant

Waste classification through intelligent methods has become a key factor for human beings to achieve sustainable development. Intelligent. Waste classification can be applied to mobile devices, intelligent recyclable trash cans, etc. It is beneficial to the environment and improves the recycling of waste resources.



## FEASIBILITY STUDY

The feasibility study is major factor, which contributes to the analysis and development of the system. The decision of the system analyst whether to design a particular system or not depends on its feasibility study.

1. Operational Feasibility:-

   o The introduction to this system is not going to hamper any user of the system.

   o The proposed system is very flexible and user friendly

   o The proposed system produces best results and gives high performance. It can be implemented easily .So this project is operationally feasible.

2. Technical Feasibility:-

   o This feasibility deals with technicality of the system.

   o No efficient manpower is required to handle the system.

3. Economic Feasibility:-

   o Economic Feasibility deals about the economic impact faced by the organization to implement a new system.

   o Economic Feasibility in this project: - The cost to conduct a full system investigation is possible.

   - There is no additional manpower requirement.
   - There is no additional cost involved in maintaining the proposed system.

## HARDWARE CONFIGURATIONS

Processor : Intel Core i3

Processor Speed : 1GHz to 2GHz

RAM : 8GB DDR4|256GB M.2SSD

SOFTWARE CONFIGURATION  Operating System:Windows.

## Software Used

1. Python 3.10-amd64 Programming Language
2. PyCharm - Version : 2022.3.3 Community edition
3. tensorflow-2.10.1
4. OpenCV - Python library - Version : 4.5.4.60
6. Pip-version-21.3.1
7. Cvzone-version-1.5.6
8. Numpy-version-1.21.6

# SYSTEM TESTING & MAINTENANCE

## SYSTEM TESTING

Software Testing Fundamentals (STF) is a platform to gain (or refresh) basic knowledge in the field of Software Testing. If we are to cliché it, the site is of the testers, by the testers, and for the testers. Our goal is to build a resourceful repository of Quality Content on Quality.

The box approach: Software testing methods are traditionally divided into white- and blackbox testing. These two approaches are used to describe the point of view that a test engineer takes when designing test cases.

## WHITE BOX TESTING

White-box testing (also known as clear box testing, glass box testing, transparent box testing and structural testing, by seeing the source code) tests internal structures or workings of a program, as opposed to the functionality exposed to the end-user. In white-box testing an internal perspective of the system, as well as programming skills, are used to design test cases. The tester chooses inputs to exercise paths through the code and determine the appropriate outputs. This is analogous to testing nodes in a circuit, e.g. in-circuit testing (ICT).

While white-box testing can be applied at the unit, integration and system levels of the software testing process, it is usually done at the unit level. It can test paths within a unit, paths between units during integration, and between subsystems during a system–level test. Though this method of test design can uncover many errors or problems, it might not detect unimplemented parts of the specification or missing requirements.

Techniques used in white-box testing include:

API testing – testing of the application using public and private APIs (application programming interfaces)

Code coverage – creating tests to satisfy some criteria of code coverage (e.g., the test designer can create tests to cause all statements in the program to be executed at least once)

## BLACK BOX TESTING

Black-box testing treats the software as a "black box", examining functionality without any knowledge of internal implementation, without seeing the source code. The testers are only aware of what the software is supposed to do, not how it does it. Black-box testing methods include: equivalence partitioning, boundary value analysis, all-pairs testing, state transition tables, decision table testing, fuzz testing, model-based testing, use case testing, exploratory testing and specification-based testing.

Specification-based testing aims to test the functionality of software according to the applicable requirements. This level of testing usually requires thorough test cases to be provided to the tester, who then can simply verify that for a given input, the output value (or behaviour), either "is" or "is not" the same as the expected value specified in the test case. Test cases are built around specifications and requirements, i.e., what the application is supposed to do. It uses external descriptions of the software, including specifications, requirements, and designs to derive test cases. These tests can be functional or nonfunctional, though usually functional.

Specification-based testing may be necessary to assure correct functionality, but it is insufficient to guard against complex or high-risk situations. One advantage of the black box technique is that no programming knowledge is required. Whatever biases the programmers may have had, the tester likely has a different set and may emphasize different areas of functionality. On the other hand, black-box testing has been said to be "like a walk in a dark labyrinth without a flashlight. “Because they do not examine the source code, there are situations when a tester writes many test cases to check something that could have been tested by only one test case, or leaves some parts of the program untested.

## GREY BOX TESTING :

Grey-box testing involves having knowledge of internal data structures and algorithms for purposes of designing tests, while executing those tests at the user, or black-box level. The tester is not required to have full access to the software's source code not in citation given] Manipulating input data and formatting output do not qualify as grey-box, because the input and output are clearly outside of the "black box" that we are calling the system under test. This distinction is particularly important when conducting integration testing between two modules of code written by two different developers, where only the interfaces are exposed for test.

However, tests that require modifying a back-end data repository such as a database or a log file does qualify as grey-box, as the user would not normally be able to change the data repository in normal production operations. Grey-box testing may also include reverse engineering to determine, for instance, boundary values or error messages.

By knowing the underlying concepts of how the software works, the tester makes betterinformed testing choices while testing the software from outside. Typically, a grey-box tester will be permitted to set up an isolated testing environment with activities such as seeding a database. The tester can observe the state of the product being tested after performing certain actions such as executing SQL statements against the database and then executing queries to ensure that the expected changes have been reflected. Grey-box testing implements intelligent test scenarios, based on limited information. This will particularly apply to data type handling, exception handling, and so on.

## SYSTEM MAINTENANCE

An integral part of software is the maintenance one, which requires an accurate maintenance plan to be prepared during the software development. It should specify how users will request modifications or report problems. The budget should include resource and cost estimates. A new decision should be addressed for the developing of every new system feature and its quality objectives. The software maintenance, which can last for 5–6 years (or even decades) after the development process, calls for an effective plan which can address the scope of software maintenance, the tailoring of the post-delivery/deployment process, the designation of who will provide maintenance, and an estimate of the life-cycle costs. The selection of proper enforcement of standards is the challenging task right from early stage of software engineering which has not got definite importance by the concerned stakeholders.

# SYSTEM DESIGN

# Python

Python is a widely used general-purpose, high level programming language. It was initially designed by Guido van Rossum in 1991 and developed by Python Software Foundation. It was mainly developed for emphasis on code readability, and its syntax allows programmers to express concepts in fewer lines of code. Python is a programming language that lets you work quickly and integrate systems more efficiently. Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

• Python is Interpreted − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.

• Python is Interactive − You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.

Python is Object-Oriented − Python supports Object-Oriented style or technique of programming that encapsulates code within objects Python's Features includes-

• Easy-to-learn − Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.

• Easy-to-read − Python code is more clearly defined and visible to the eyes.

• Easy-to-maintain − Python's source code is fairly easy-to maintain

. • A broad standard library − Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.

• Interactive Mode − Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.

• Portable − Python can run on a wide variety of hardware platforms and has the same interface on all platforms.

• Extendable − You can add low-level modules to the Python interpreter. These modules enable pro

grammers to add to or customise their tools to be more efficient.

• Databases − Python provides interfaces to all major commercial databases.

• GUI Programming − Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.

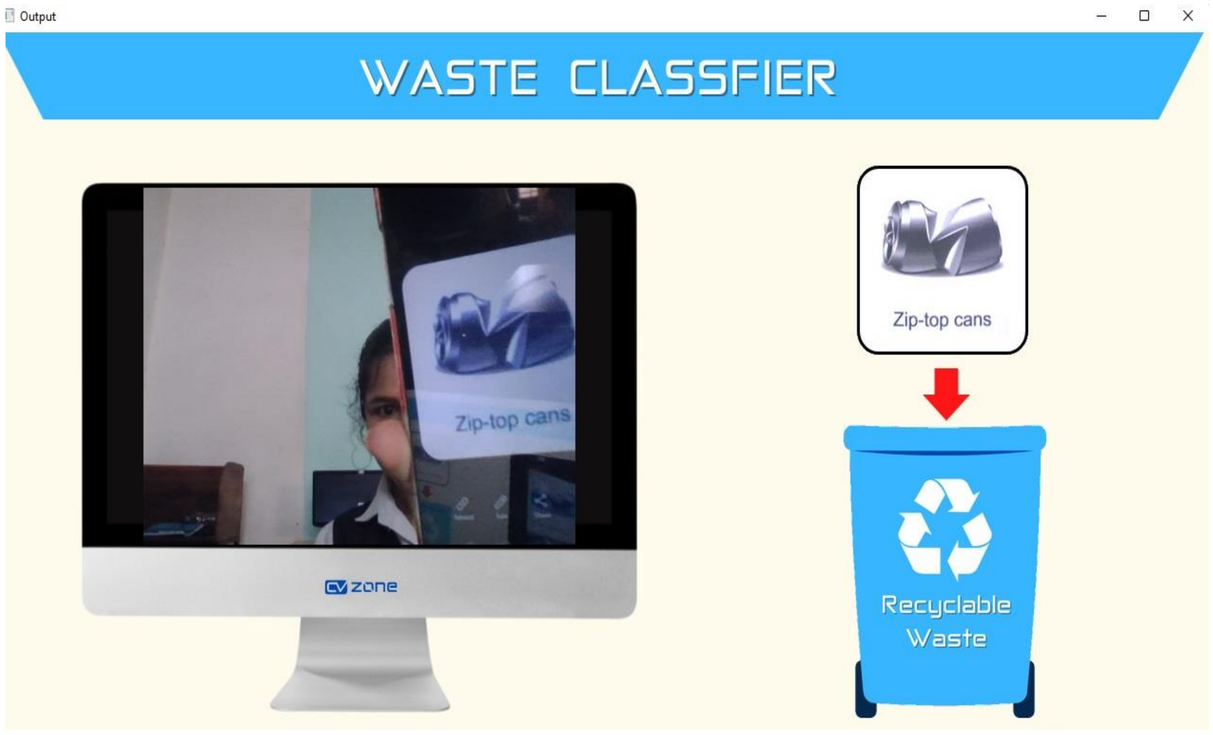
## PyCharm

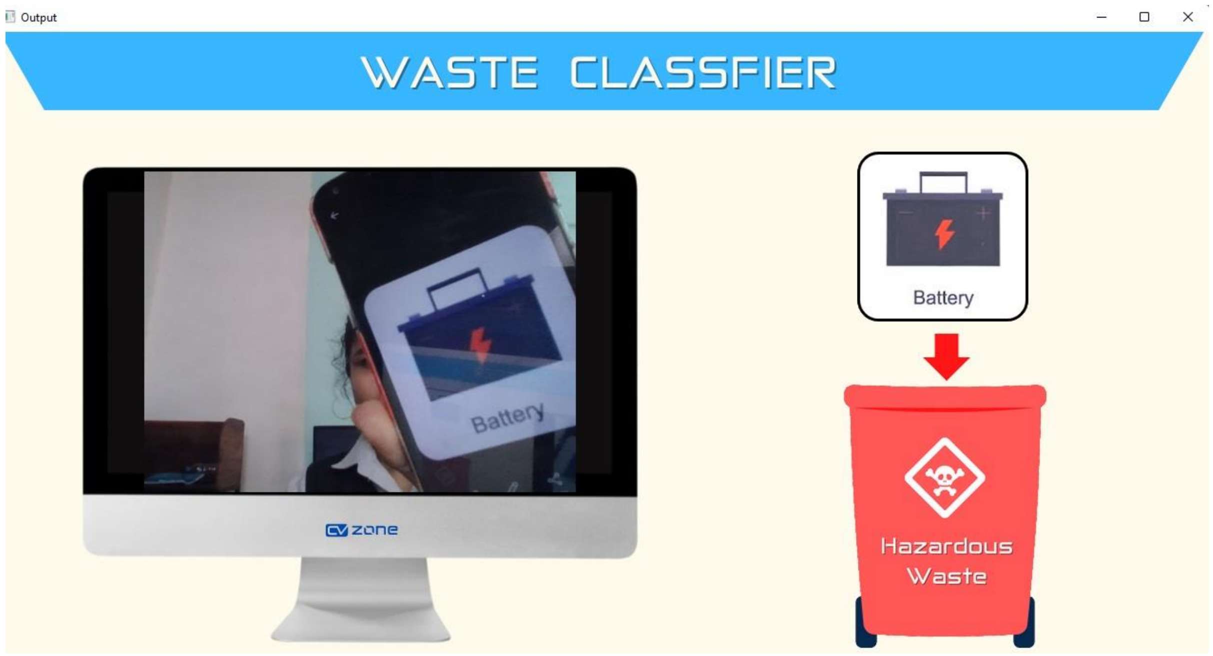
PyCharm is an integrated development environment (IDE) used in computer programming, specifically for the Python language. It is developed by the Czech company JetBrains.PyCharm Community Edition is totally free and open-source, available under the Apache 2.0 license. PyCharm makes it easier for programmers to write various web applications in Python supporting widely used web technologies like HTML, CSS, JavaScript, TypeScript and CoffeeScript PyCharm is available in three editions: Professional, Community, and Educational (Edu). The Community and Edu editions are open-source projects and they are free, but they have less features. PyCharm Edu provides courses and helps you learn programming with Python.

## TensorFlow

TensorFlow APIs are arranged hierarchically, with the high-level APIs built on the low-level APIs. Machine learning researchers use the low-level APIs to create and explore new machine learning algorithms. In this class, you will use a high-level API named tf.keras to define and train machine learning models and to make predictions. tf.keras is the TensorFlow variant of the open-source Keras API. TensorFlow is a free and open-source software library for machine learning and artificial intelligence. It can be used across a range of tasks but has a particular focus on training and inference of deep neural networks. TensorFlow was developed by the Google Brain team for internal Google use in research and production. The initial version was released under the Apache License 2.0 in 2015.[1][8] Google released the updated version of TensorFlow, named TensorFlow 2.0, in September 2019. TensorFlow can be used in a wide variety of programming languages, including Python, JavaScript, C++, and Java. This flexibility lends itself to a range of applications in many different sectors.

Screenshotes





# CONCLUSIONS

Deep learning techniques have shown promising results for waste classification tasks. These techniques can accurately classify different types of waste, including plastic, paper, and metal, and can also identify specific subcategories within these broad classes. Deep learning algorithms can be trained on large datasets of images or other types of data, allowing them to learn complex patterns and features that are difficult to identify using traditional methods. Overall, the use of deep learning techniques can improve the efficiency and accuracy of waste classification systems, making it easier to properly manage and recycle different types of waste. The main objective of this study is to study the properties of solid waste some of the intelligent methods used to classify waste, including deep learning, were mentioned and analyzed Several of the model techniques for managing waste classification in daily life. Intelligent trash classification is essential because of the growing amount of waste. We reviewed the most popular smart methods used for waste classification and make a comparision between them.

## Drawbacks

- This application not suitable for inputting large amounts of data,not very accurate.
- Expensive

# REFERENCES

1. Feng, J.-w., and Tang, X.-y.: ‘Office garbage intelligent classification based on inception-v3 transfer learning model’, in Editor (Ed.)(Eds.): ‘Book Office garbage intelligent classification.
2. Costa, B.S., Bernardes, A.C., Pereira, J.V., Zampa, V.H., Pereira, V.A., Matos, G.F., Soares, E.A., Soares, C.L., and Silva, A.F.: ‘Artificial intelligence in automated sorting in trash recycling’, in Editor (Ed.)^(Eds.): ‘Book Artificial intelligence in automated sorting in trash recycling’ (SBC, 2018, edn.), pp. 198-205
3. Xu, X., Qi, X., and Diao, X.: ‘Reach on waste classification and identification by transfer learning and lightweight neural network

## Code

```
import os

import cvzone

from cvzone.ClassificationModule import Classifier

import cv2

cap = cv2.VideoCapture(2)

classifier = Classifier('Resources/Model/keras_model.h5',
'Resources/Model/labels.txt')

imgArrow = cv2.imread('Resources/arrow.png', cv2.IMREAD_UNCHANGED)

classIDBin = 0

# Import all the waste images

imgWasteList = []

pathFolderWaste = "Resources/Waste"

pathList = os.listdir(pathFolderWaste)

for path in pathList:

    imgWasteList.append(cv2.imread(os.path.join(pathFolderWaste, path),
cv2.IMREAD_UNCHANGED))


# Import all the waste images

imgBinsList = []

pathFolderBins = "Resources/Bins"

pathList = os.listdir(pathFolderBins)

for path in pathList:

    imgBinsList.append(cv2.imread(os.path.join(pathFolderBins, path),
cv2.IMREAD_UNCHANGED))
```

```
# 0 = Recyclable
# 1 = Hazardous
# 2 = Food
# 3 = Residual

classDic = {0: None,
            1: 0,
            2: 0,
            3: 3,
            4: 3,
            5: 1,
            6: 1,
            7: 2,
            8: 2}

while True:
    _, img = cap.read()
    imgResize = cv2.resize(img, (454, 340))

    imgBackground = cv2.imread('Resources/background.png')

    predection = classifier.getPrediction(img)

    classID = predection[1]
    print(classID)
    if classID != 0:
        imgBackground = cvzone.overlayPNG(imgBackground,
imgWasteList[classID - 1], (909, 127))

        imgBackground = cvzone.overlayPNG(imgBackground, imgArrow, (978,
320))

   classIDBin = classDic[classID]

   imgBackground = cvzone.overlayPNG(imgBackground,
imgBinsList[classIDBin], (895, 374))

  imgBackground[148:148 + 340, 159:159 + 454] = imgResize

    # Displays

    # cv2.imshow("Image", img)

    cv2.imshow("Output", imgBackground)

    cv2.waitKey(1)
```