



INSTITUTO
UNIVERSITÁRIO
DE LISBOA

Detection of garbage outside of the deposition equipment: a study on
classification-based and object detection-based computer vision approaches

Gonçalo Filipe Constantino Soares

Master in Telecommunications and Computer Engineering

Supervisor:

Phd Doctor Tomás Gomes da Silva Serpa Brandão, Assistant Professor
ISCTE - Instituto Universitário de Lisboa

Co-Supervisor: Phd Doctor João Carlos Ferreira, Assistant Professor with
Habilitation

ISCTE - Instituto Universitário de Lisboa

October, 2023



TECHNOLOGY
AND ARCHITECTURE

Department of Information Science and Technology

Detection of garbage outside of the deposition equipment: a study on
classification-based and object detection-based computer vision approaches

Gonçalo Filipe Constantino Soares

Master in Telecommunications and Computer Engineering

Supervisor:

Phd Doctor Tomás Gomes da Silva Serpa Brandão, Assistant Professor
ISCTE - Instituto Universitário de Lisboa

Co-Supervisor:

Phd Doctor João Carlos Ferreira, Assistant Professor
ISCTE - Instituto Universitário de Lisboa

October, 2023

Acknowledgements

I would like to thank my family, particularly my parents for the education they gave me and the incentive they gave me to finish this dissertation. Without them it would undoubtedly not have been possible.

I would also like to thank my supervisors Dr. Tomás Brandão and Dr. João Ferreira for their guidance and help in completing this dissertation. Without their mentorship and time, this dissertation would not have been possible.

Gonçalo Soares

Resumo

O excesso de acumulação de lixo é um problema em grandes cidades onde a produção de resíduos urbanos é elevada. Este problema leva a que as equipas de recolha de lixo realizem um maior esforço para combater tal situação. Sendo assim, nesta dissertação é proposto dois sistemas de identificação de lixo, que serão comparadas, para solucionar este problema na capital de Portugal. O objetivo principal desta proposta é facilitar o trabalho de recolha de resíduos na cidade de Lisboa, trabalho este realizado pelas equipas dos Centros de Recolha de Resíduos de Lisboa. Com o intuito de facilitar e ajudar a coleta de resíduos, a Câmara Municipal de Lisboa colaborou com os inspetores da equipe de coleta e criou a "LxDataLab", uma plataforma que disponibiliza uma variedade de datasets. As fotos são tiradas de câmaras fotográficas de smartphones pelas equipes de recolha e normalmente são tiradas de veículos em movimento ou até mesmo de residentes locais. O processamento das imagens é realizado de forma diferente em ambos os sistemas criados. Um dos sistemas utiliza as imagens originais, muda a sua resolução e reparte a imagens em várias sub-imagens que contêm uma porção da imagem alterada. Neste sistema é usado redes neuronais feitas e à mão e outras pré-treinadas e diferentes métodos para obter os resultados usando o dataset das sub-imagens. Enquanto que no outro sistema, usa as imagens originais e faz a sua avaliação usando um algoritmo chamado de yolov5. Por fim, é feito uma comparação justa entre os dois sistemas para determinar qual é o mais eficaz avaliando os valores de precisão e loss.

Keywords: Redes neuronais, resíduos, visão por computador, identificação, lixo, aprendizagem automática.

Abstract

Excessive waste accumulation is a problem in large cities and capitals where urban waste production is high. This problem leads waste collection teams to make a greater effort to combat the situation. Therefore, this dissertation proposes two waste identification systems, which will be compared, to solve this problem in Portugal's capital. The main objective of this proposal is to facilitate the work of waste collection in the city of Lisbon, which is carried out by the teams of the Lisbon Waste Collection Centers. In order to facilitate and help waste collection, Lisbon City Council collaborated with the collection team inspectors and created "LxDataLab", a platform that provides a variety of datasets. The images are taken from smartphone cameras by the collection crews and are usually taken from moving vehicles or even local residents. Image processing is carried out differently in the two systems created. The patch-based garbage detector system uses the original images, changes their resolution and splits the image into several sub-images that contain a portion of the altered image. This system uses hand-made, pre-trained neural networks and different methods to obtain the results using the dataset of sub-images. The other system, called the object detection-based system, uses the original images and evaluates them using an algorithm called yolov5. Finally, a fair comparison is made between the two systems to determine which is the most effective by evaluating the accuracy and loss values.

Keywords: Neural networks, waste, computer vision, identification, garbage, machine learning.

Contents

Acknowledgements	iii
Resumo	v
Abstract	vii
List of Figures	xi
List of Tables	xiii
Chapter 1.	1
1.1. Introduction	1
1.2. Motivation	2
1.3. Research Questions	4
1.4. Objectives	5
1.5. Research Methodology and Structure of the Dissertation	5
Chapter 2. Concepts and Literature Review	9
2.1. Concepts	9
2.2. Related work	13
2.3. Summary	17
Chapter 3. Garbage Detection system	19
3.1. Garbage Detection Approaches	19
3.2. Dataset	21
Chapter 4. Experimental results	33
4.1. General experimental setup	33
4.2. Patch-based approach	34
4.3. Object detection-based approach	41
	ix

Chapter 5. Conclusion	49
References	51

List of Figures

1	Design Science Research Methodology (Adapted from [3]).....	6
2	CNN architecture.....	9
3	Data Augmentantion.....	11
4	Yolov5 architecture.....	12
5	General System Architecture.....	21
6	Garbage detection system behavior.....	22
7	Example of YOLO classification.....	23
8	Data built.....	23
9	Data acquisition.....	24
10	Images taken by Collection Garbage Inspectors.....	25
11	Labelling software.....	25
12	Example of an .xml file with the bounding box coordinates.....	26
13	Yolov5 annotations.....	27
14	Factor value calculation.....	28
15	New Bounding boxes values.....	29
16	New Annotation.....	29
17	IoU.....	30
18	IoU Further Demonstration.....	30
19	Accuracy plot using Data Augmentation.....	35
20	Loss plot using Data Augmentation.....	36

21 Precision values example.....	42
22 Precision vs recall curve explained.....	44
23 Precision curve run 6.....	45
24 Examples for the test results of the 6° run.....	45
25 4 examples for the test results lower then 0.5.....	46
26 Patch-based garbage detector black and white image distinguish.....	47

List of Tables

1	Simple 2 convolutional layers CNN: results for 60/10/30 dataset split.....	36
2	Simple 2 convolutional layers CNN: results for 70/10/20 dataset split.....	36
3	Simple 4 convolutional layers CNN: results for 60/10/30 dataset split.....	37
4	Simple 4 convolutional layers CNN: results for 70/10/20 dataset split.....	37
5	MobileNet results 60/10/30.....	38
6	MoibileNet results 70/10/20.....	39
7	ResNet50 results 60/10/30.....	39
8	ResNet50 results 70/10/20.....	40
9	DenseNet results 60/10/30.....	40
10	DenseNet results 70/10/20.....	40
11	Object detection-based experiences settings.....	41
12	Runs precision results.....	44
13	Both systems overall results.....	47

CHAPTER 1

1.1. Introduction

Product leftovers are considered urban solid waste, commonly known as garbage. Recyclable products, organic waste, garden waste, and bulky waste all fall under this category. Its management has been one of Portugal's most significant issues, particularly for towns and government officials.

In recent years, global population growth, together with a consumerist society, has resulted in more production, more consumption, and consequently a greater volume of produced garbage, resulting in insufficient infrastructure for waste collection and disposal, causing significant environmental harm.

According to the information published by the Portuguese Environment Agency (APA)¹, in 2021, the total waste production in mainland Portugal was, approximately, 5.31 million tons, 0.04 more compared to 2020. These values mean that each Portuguese inhabitant produces an average of about 511 kg of garbage per year or 1.4 kg per day.

Efforts to reduce the above mentioned numbers rely on raising the proportion of recyclable trash, ensuring the economic viability of waste-generating models, and reducing the amount of garbage disposed of in landfills. Much of the produced waste, particularly solid urban waste, is recyclable, meaning that all of the waste collected goes through a process that turns wasted materials into new goods. Different recycling techniques are used depending on the type of trash, therefore using methods that allow for proper garbage disposal in the appropriate equipment might be beneficial. Existing techniques for trash separation, notably selective sorting (ecopoints, glass), and a series of awareness campaigns to make collection job easier have all become necessary, but they are still insufficient to result in significant environmental gains.

¹<https://apambiente.pt/residuos/dados-sobre-residuos-urbanos>

According to APA¹, about 78% of the garbage collected in 2021 is from the category of urban waste and it is this trash category that causes the most accumulation placed outside of the containers. Citizens place their garbage outside of the containers due to excessive waste generation or waste disposal facilities near their homes, hence automatic detection of such situations might aid the collection process.

Lisbon City Hall is pursuing a number of methods to address this issue, including:

- The installation of subsurface recycling equipment in the hopes of reducing the visual impact that rubbish has on the streets. This sort of equipment comprises large waste containers, which people frequently throw garbage around when they become full.
- Waste collection circuits optimization, using measures such as the installation of 1500 sensors² in many containers throughout the city - this strategy attempts to determine how full the containers are. The usage of these sensors, on the other hand, does not provide information on the accumulation of rubbish around the equipment.

Garbage disposal around equipment is common in city locations where waste disposal is high, meaning a greater effort on the part of the collection teams assigned to those places. In this regard, it is critical to plan ahead of time and anticipate scenarios. There is room for improvement in various city regions with regard to the handling of garbage collection in Lisbon. In this context, the development of a trained model to detect and classify waste dumped outside of disposal equipment can help to improve collection operations management. Thus, the creation of a computer vision-based algorithm for identifying residues in images is the main goal of this Dissertation research, where systems capable of detecting waste through image recognition were implemented using deep learning principles.

1.2. Motivation

The problem addressed in this dissertation is was originally studied by former student Soraia Hermínia Fernandes [1]. Her work was presented at the end of November 2021

²<https://lisboainteligente.cm-lisboa.pt/lxi-iniciativas/sensorizacao-dos-depositos-coletivos-de-residuos/>

and it reached an accuracy of 84% to identify the rubbish outside the containers. This dissertation is an extension of that work, with the aim of achieving results with higher precision and loss values and less false positive results.

One of the biggest problems in the dataset is the diversity of resolution and aspect ratios in the source images. Images were collected by different entities, using different cameras with different qualities. The image resolutions are not uniform, making it difficult for the algorithm proposed in [1] to detect the garbage outside the containers. Another problem that was also mentioned in [1], was the small number of images available, which posed challenges to train the garbage identification system. It is noteworthy to mention the idea explored in the algorithm created by the former colleague, which consisted in splitting the source images into blocks of 64 by 64 pixels, and then classifying each block according to the presence of trash (or not) using Convolutional Neural Networks (CNN). This idea allowed to obtain a larger number of (small) images for CNN training. However, depending on the resolution of the source image before splitting, the scale of the content associated to each image block could vary significantly.

In areas where urban waste consumption is high, the problem of improper waste disposal is frequent, causing waste to be placed outside the bins. In addition, large volumes of waste, such as furniture or household utensils, are not properly recycled, often due to a lack of knowledge on the part of citizens about where and how to recycle such waste. This type of problem often emerge in large cities and capitals, where waste consumption is high. For this reason, the proposal in this dissertation seeks to solve this problem by exploring two waste identification systems, one of which will be based on the idea proposed in [1].

Lisbon City Council worked with the collection team inspectors to develop “LxData-Lab”, a platform that offers a variety of datasets, in order to facilitate and aid garbage collection. The collection teams use smartphone cameras to capture the photographs, which are typically taken from moving vehicles or even by residents. In the system based on [1], the dataset images will be divided into small sub-images of 32 by 32 pixels, instead of 64 by 64, in order to also obtain a larger dataset. While the other system use the original images without any alterations being made

Given the advances in technology, there are several systems and ways of classifying and detecting various objects. As mentioned earlier, one of the systems is based [1] using deep learning applications based on convolutional neural networks that have proven to be effectively used for accurate image classification.

In addition, the other system to be explored is based on object detection and uses the yolov5 algorithm ³, which detects and recognizes objects in real time. It locates a region of interest in the image and classifies this region in the same way that a standard image classifier would. Multiple regions of interest locating different types of objects can be present in a single image. This elevates object detection to a more complex image classification issue.

While the first system based in [1] focuses on detecting where garbage is and provides is class label, the other aims on assigning labels to images or regions. However, both have the same goal to take an image and autonomously knowing whether or not it contains garbage in the image. These types of technology could help waste collection teams to reduce excess waste in cities where this problem is most prominent.

1.3. Research Questions

No evaluation will ever be better than human evaluation, however, with the passage of time and advances in the areas of computer vision and artificial intelligence, it is possible to detect and classify objects almost as perfectly as humans. In this way, by creating two systems that use different computer vision methods, it will be possible to evaluate the same fact. As a result, the following questions will be addressed in this work:

- Q1 - Which leads to better results in identifying garbage - an approach based on classifying blocks of images (similar to [1]), or an approach based on object detection?
- Q2- Is it worth to preprocess the dataset images in order to “normalize” their resolutions, or not?

³<https://blog.roboflow.com/yolov5-improvements-and-evaluation/>

1.4. Objectives

The focus of this dissertation is to create two garbage detection systems and make a fair comparison between the two to determine which of the two approaches achieves the best results. The developed work was expected to achieve the following goals:

- To obtain a larger and organized dataset, suitable for training and comparing both garbage detection approaches;
- To train and test CNN-based models built from scratch as well pre-trained CNN network models using transfer learning;
- To use the yolov5 algorithm for the detection of garbage using single and multiple class approaches, compare these approaches and assess the viability of multiple garbage classes;
- To determine which of the proposed systems leads to the best garbage detection accuracy;

1.5. Research Methodology and Structure of the Dissertation

Since the main goal of this dissertation is to create two systems that help identify waste outside the containers, this study's methodology is based on the Design Science Research (DSR) concept. This methodology is appropriate for tackling real-world problems and is geared at artefact creation [2]. As shown in Figure 1, the DSR model defines a series of fundamental phases that will lead to the creation of a final artefact.

After identifying the problem, the first stage of the iterative process is to define the objectives, which leads to the formation of research questions that will be answered with the completion of this dissertation. This step was addressed in sections 1.2 to 1.4 of this dissertation.

Therefore, the first chapter corresponds to this stage. This chapter covers the dissertation's topic, motivation, research questions, and objectives, as well as the research technique model that was used.

Chapter 2 provides an overview of the main deep learning ideas, with a focus on convolutional neural networks and yolov5 algorithm, in order to better understand the material of the next chapters. This chapter also includes a literature review, which is a

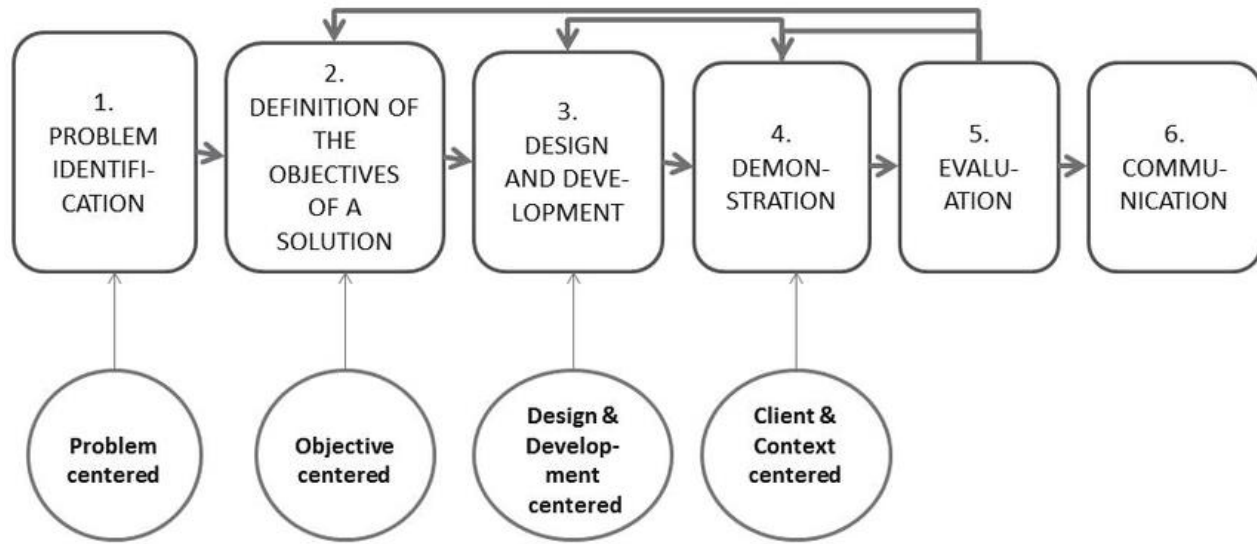


Figure 1. Design Science Research Methodology (Adapted from [3])

brief summary of the related work that has already been done on the topic. Finally, an overview of the most recent state-of-the-art research is performed and related to the study's topic.

The artefact is defined in the next stage, which is design and development. In the context of this work, the artefact is a convolutional neural network-based garbage detection algorithm and yolov5 algorithm. It will be created in an iterative manner, comparable to the agile software development technique [4].

Chapter 3 deals precisely with the developments carried out in both systems. In chapter 3, the both suggested system's to detect residues outside of designated equipment is described, followed by a cogent analysis of the problem to be solved and how the functional prototype developed can answer the research objectives. In addition, it will be explained how the datasets were obtained and how the annotations/labels used in each system were made. Changes to the format of the dataset images will also be explained in order to solve the research objectives.

The verification of the model's reliability is then put into practice during the demonstration step. It goes into great detail and explains how the model is trained to detect and classify things in images. Preliminary results are expected to be produced in this stage, in addition to the demonstration of test experiments or simulations.

Therefore, chapter 4 covers this stage by showing the experiments carried out and the results obtained. This chapter reveals the results of both systems and the different methods and configurations used in each experiment. It will be revealed which programs and packages were used for the use of the algorithms and finally a fair comparison of the two systems will be made in order to know which is the most effective in terms of accuracy.

The artefact's usability and value are demonstrated in the last step of communication, which includes writing a dissertation.

The finished artefact there should be two garbage identification systems, one using neural networks and the other using the yolov5 algorithm. Both being two categorization systems that can recognize garbage.

Finally, in the fifth and final chapter covers the finished artifact. The dissertation's key conclusions are drawn and topics for future research are suggested. Besides that, the research questions in section 1.3 will be answered.

CHAPTER 2

Concepts and Literature Review

This chapter is organized into three sections. It starts by showing important theoretical concepts used on this project. The second section focuses on related work that addresses various machine learning applications in the trash management context. Finally, the last section summarizes the literature search and presents possible contributions of this work.

2.1. Concepts

2.1.1. Convolutional Neural Network (CNN)

A Convolutional Neural Network is an artificial neural network with an architecture that is designed to learn hierarchies of spatial features, typically applied to images. CNN's are Deep Learning neural networks whose architecture includes multiple layers and learns from large amounts of data.

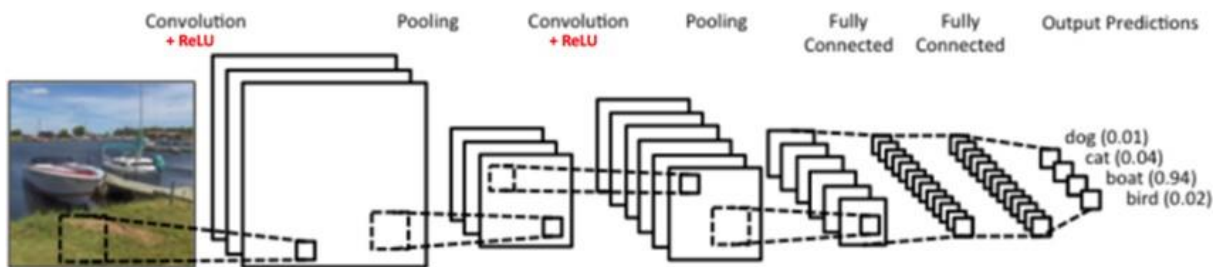


Figure 2. CNN architecture⁴

Figure 2 depicts the general architecture of a CNN. As can be observed from the figure, it consists of different layer types such as:

⁴<https://matheusfacure.github.io/2017/03/12/cnn-captcha/>

- Convolutional layers - which apply convolution operations between a filter core and its input data matrix. It is in the convolution process that the filter coefficients (neural network weights) are determined.
- Pooling layers - which are used to reduce the size of the matrices, simplifying the information at the output of the convolutional layers.
- Fully connected or Dense layers - which connected the convolutional part of the network to the networks output layer, which performs class predictions for the input image. All neurons of a Dense layer are connected to all neurons in the previous layer.

It's important to understand what a neuron is in this context. It is basically a unit that computes a weighted sum of values presented at its input connections. The output of the neuron will be the result of applying an activation function to the weighted sum. Each layer is typically composed of several neurons (which may be thousands depending on the developed network). Each neurons' input is typically connected to an output of the previous layer and the neurons' output is connected to the next layer.

In the example illustrated in figure 2, the image of a boat is subject to classification. After going through all the layers mentioned above, the predictions are computed resulting in 94% confidence score that the input image is indeed representing a boat.

One of the obstacles when working with CNN and other neural networks is overfitting. When a model learns not only the information, but also the noise on the input image data it may not generalize well, leading to an increased number of incorrect classifications on new image data.

One possible solution to mitigate this problem is the use of *dropout*. It consists of 'turning off' a random set of neurons at the beginning of each iteration of the training process. As a result, each neuron is forced to learn more robust features that will be useful for classifying the image.

2.1.2. Data Augmentation

Data augmentation, which is a process widely used for CNN training, consists of building different versions of the same image using shift, flip, zoom and rotation operations,

among others. This strategy allows the network to rely on relevant information while avoiding overfitting secondary details. Figure 3 shows an image subject to different augmentation techniques.

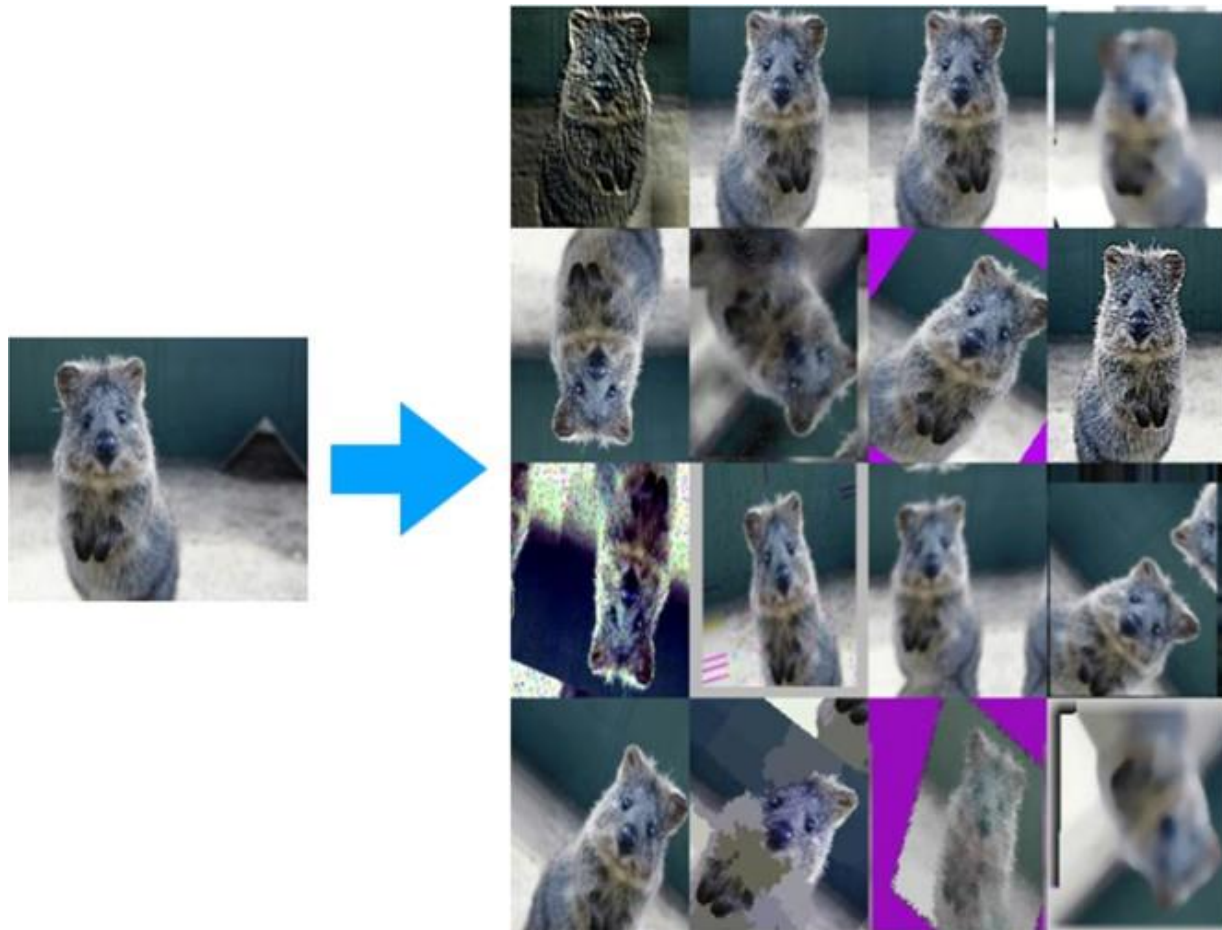


Figure 3. Data Augmentation⁵

2.1.3. Transfer Learning

Due to the complexity of the training procedures, training a CNN from scratch is a time-consuming and demanding process in terms of computational memory and power. Transfer Learning is a technique that aims to improve traditional machine learning by using knowledge from one or more tasks in the original network to access and improve the learning of the new network.

⁵<https://hackernoon.com/a-gentle-introduction-to-data-augmentation>

2.1.4. Object Detection

Object detection consists of showing precisely where the desired object is located with maximum precision using bounding boxes but also provides class labels. Classification, on the other hand, focuses on labeling images or regions. High performance object detection can be accomplished using YOLO models. YOLO creates a grid structure out of an image, and each grid finds objects inside it. Based on the data streams, they can be used for real-time object detection. The YOLO model used in this project is yolov5.

Figure 4 shows a high-level object detection architecture that reveals how yolov5 has improved speed and design.

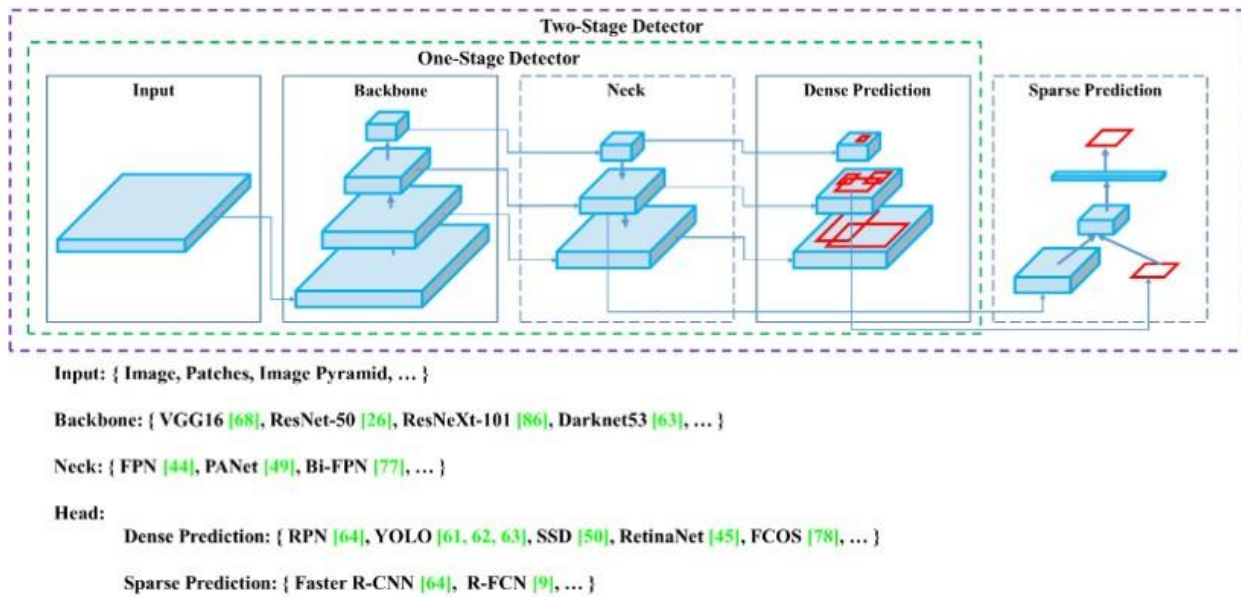


Figure 4. Yolov5 architecture⁶

A typical object detector consists of an head, that predicts classes and bounding boxes, and a backbone that can use a pre-trained CNN. The backbones can operate on platforms with a Graphic Processing Unit (GPU) or a Central Process Unit (CPU). The head can be either one-stage (e.g., YOLO, SSD, RetinaNet) or two-stage (e.g. Faster R-CNN) object detector for the sparse prediction. Modern object detectors have a layer that collects feature maps (the neck) between the backbone and the head.

⁶<https://medium.com/analytics-vidhya/object-detection-algorithm-yolo-v5-architecture-89e0a35472ef>

A backbone and Spatial Pyramid Pooling (SPP) block, respectively, are used in yolov4 to increase the receptive field, isolate the important characteristics, and maintain the network's operating speed. For parameter aggregation from various backbone levels, Path Aggregation Network for Instance Segmentation (PAN) is applied.

All concepts mention above are used and important to understand the methods used to approach the problem of identify garbage outside the containers.

2.2. Related work

2.2.1. Systematic Literature review

The process described by Briner and Denyer in [5], as well as the characteristics defined in the PRISMA statement in [6] were followed in order to perform a succinct systematic review on the recognition of waste outside the disposal equipment using computer vision tools.

The review's methodological approach is divided into three stages. The goals and needs of the revision are determined in the first stage, after which a proposal for revision is made and criteria to support the revision are developed. The second stage is concerned with research, quality assessment, data gathering, and data analysis. The last stage is reporting the review's findings.

During the months of November and January 2021/2022, a systematic literature search on the subject of detecting garbage photos using computer vision techniques was conducted. The terms "waste", "computer", "vision", "identification", "neural networks", "bins", "classification", "machine learning" and "garbage" were searched in all publications in the Scopus databases to discover scholarly articles.

The following sub-sections describe the related work on the use of machine learning techniques for trash recognition and classification.

2.2.2. Convolutional Neural Networks on garbage-related tasks

Quantifying littering is a crucial step in enhancing city cleanliness. When human interpretation is too laborious or, in certain situations, impossible, an objective cleanliness index could help prevent littering by raising awareness and encouraging appropriate

behavior. In this article [7], a completely automated computer vision program for quantifying littering using images from the sidewalks and streets was introduced. A deep learning-based framework was utilized to locate and classify different forms of garbage. Since there was no waste dataset available, an acquisition mechanism that is fitted on a vehicle has been developed. Collected images of different waste products gathering pictures of various trash products. The constructed system is then trained on these images, and its performance is benchmarked. OverFeat-GoogLeNet model was used and presented by [8], which is a type of CNN. The original edition of OverFeat involves use of AlexNet-based picture representation [9].

High computational costs are another frequent problem in image classification, which frequently cause long development times and large prediction model sizes. It is crucial in this domain to have a lightweight model that is very accurate and transparent about its process. Nonso Nnamoko, Joseph Barrowclo and Jack Procter on their paper [10] investigate this issue by exploring two image resolution sizes (i.e., 225 264 and 80 45) to compare the performance of their custom five-layer convolutional neural network in terms of development time, model size, predictive accuracy, and cross-entropy loss in order to evaluate the issue of computational cost. Their hypothesis is that a model trained with lower image resolution will have a lighter weight and/or comparable accuracy to a model trained with higher image resolution. A random guess classifier to compare the outcomes in the absence of trustworthy baseline research to compare the accuracy and loss of the bespoke convolutional network was trained. The findings demonstrate that low image resolution results in a lighter model with shorter training times, and the accuracy obtained (80.88%) is higher than that obtained by the larger model (76.19%).

For this paper [11], CNN was used along with hardware and software and a cloud server to sort the rubbish by categories with a smart bin. In this case the division is done by categories of household waste: recyclable waste, hazardous waste, kitchen waste and other waste. The hardware module is in charge of recognising and photographing the input rubbish via the front-end sensors, then recording and uploading the images to the cloud server. Garbage is classified, collected, and processed in accordance with the instructions supplied by the cloud server. It detects overflowing garbage bins, sends the

associated data to the cloud server for analysis and processing, and automatically ejects the collecting bin in accordance with the manager's instructions

2.2.3. Object detection based garbage detection

The work presented in [12] is the result of a preliminary research that aims to use computer vision techniques to replace the vision techniques to replace the current method of waste container identification via radio frequency identification. Compared to the current method, this approach is more agile and reduces the resources required for implementation. A approach discussed here is centered on the use of convolutional neural networks, specifically the You Only Look Once (YOLO) network. Basically, in this project is created a dataset with images of garbage containers in the streets and develop an algorithm and train a YOLO neural network. The objective of the algorithm is to identify where the containers are and what type of material these containers correspond to, namely, cardboard, glass, plastic and unreferenced. In [13] the same reasoning is applied, but in addition a vector of locally aggregated descriptors (VLAD) is used. However, this approach did not achieve the desired results and therefore YOLO was used. It has not been mentioned yet but in this project it is also identify in images or videos the types of containers by referring to them using labels.

For this work,[14], region based convolutional neural networks (R-CNN) are used to identify the different types of garbage in food trays. The dataset used has 1002 images captured by different smartphones. A multi-label is used to refer to what type of garbage it is, for example, if it is a napkin there is a label saying that it is 'paper_napkin'. This annotation is multi-label because the labels can be rearranged into two groups of classes, either by form or by substance, which was use to take advantage in object recognition and classification. The annotation can be structured in a multi-label configuration, with each object being tagged in terms of shape, material, and bounding box. The material label can have values from the following categories: glass, paper, metal, and plastic. The cup, plate, box, tray, cutlery, mixed trash, bottle, paper, can, and plastic shape label can take values from the set. In total there are 19 different classes and 7200 labels.

In [15], a robot was created to gather rubbish that is on the ground while employing a camera to take photographs for subsequent processing. For categorization, pre-trained convolutional networks are employed, notably MobileNetV1 with Single Shots Detectors (SSD).

Particular care should be given to the trash that has been left outside, whether it is on public city streets or in rural or suburban regions. Abandoned trash can lead to pollution and have a detrimental effect on the quality of life for locals in addition to degrading the land. In [16], B. Carolis et al. focus on creating software that can instantly analyse video feeds to find and notify the presence of abandoned garbage. For garbage identification and recognition, a modified YOLOv3 network model was used. On the dataset gathered for this purpose, the network has been refined. The findings indicate that the suggested strategy may significantly improve trash management in smart cities. In this paper, the labelling software was used to annotate each image.

How to increase the intelligence level of urban environment monitoring and evaluation has emerged as a significant research topic with the growth of smart cities in major cities both domestically and overseas, particularly the management of smart cities. In the use of intelligent urban management, it is extremely valuable to quickly and precisely identify trash from urban images. The goal set by Y. Wang and X. Zhang for in [17] is to use deep learning to automatically detect garbage. The review trash detection results on garbage photos after training a Faster R-CNN open source framework using region proposal network and ResNet network algorithm. In addition, a data fusion and augmentation strategy was suggested to increase the method's accuracy.

To create tools for detecting trash, many machine learning approaches have been investigated. These efforts help research, citizen science, and volunteer clean-up activities. Modern CNN architectures (such as Faster RCNN, Mask-RCNN, EfficientDet, RetinaNet, and YOLO-v5), two datasets of litter images, and a smartphone were used in the comparative investigation in [18]. The experimental findings show that YOLO-based object detectors have superior performance in terms of detection accuracy, processing speed, and memory footprint, making them suitable for the development of litter detection solutions.

2.3. Summary

With this literature survey, it can be concluded that the idea of a Yolo network may prove to be a promising idea to identify garbage outside the bins. There are several articles and thesis that identify garbage in the streets, on the seashore, and even garbage cans that identify whether the container is full or not. All this using CNN and R-CNNs, among other technologies. Thus, this thesis will certainly contribute to reduce and prevent pollution in cities by experimenting with various CNN models and exploring the promising hypothesis of yolo networks to identify such a problem.

CHAPTER 3

Garbage Detection system

This chapter describes two different systems for garbage detection. One is based on the system proposed in the dissertation by S. Fernandes in [1], which is based on classifying image blocks using CNNs. The other is based on detecting objects whose objective is not to classify images but to identify and locate objects, where the “object” is garbage placed outside the containers. These approaches will be designated as “patch-based” and “object-based” garbage detectors, respectively.

This chapter will also presents the necessary requirements for the realization of those systems, namely the acquisition, processing and annotation of the images used for training and testing datasets.

3.1. Garbage Detection Approaches

For this dissertation, its important to develop a system based on a supervised learning algorithm with the ability to detect garbage outside of the disposal equipment on the submitted images. It should also be able to provide the location of image parts where trash is present.

This task is accomplished differently on both systems. In the patch-based garbage detector, the image is cropped into sub-images of equal size, and each sub-image is annotated according to the presence of garbage covering (or not) the majority of its content.

On the other hand, the object-based detection system locates and identifies garbage using a bounding box and corresponding degree of confidence.

The next sections provide additional details and explanations for both approaches.

3.1.1. Patch-based garbage detector

The patch-based garbage detector consists of using CNNs and pre-trained networks to obtain results. The solution comprises feeding the categorization algorithm with images that have been taken by collection inspectors and uploaded by users of the app “A minha Rua”. For the patch-based garbage detector, each image is split into smaller blocks (patches) and each block is independently classified, rather than classifying the image as a whole. Each image block or patch must therefore be categorised as trash or not. The system’s goal is therefore to detect all trash blocks present in the image.

Splitting the image into smaller blocks also has the advantage to result in a dataset containing a larger number of images (the patches), which is more suitable for training and evaluating a CNN-based classification algorithm.

As a compromise between the complexity of implementation, the time needed for training, and the anticipated results, it was decided to start with a simple architecture. This is because during the research of works related to the topic under study, no deep learning algorithms were found that dealt directly with the recognition of residues outside of the equipment.

Besides a simple architecture, pre-trained neural networks were used, namely ResNet50, MobileNet and DenseNet in order to explore new possibilities and results.

These network architectures were tested due to the fact that they are all included in Keras applications and that they perform well when used too generic picture classification issues.

Figure 5 shows us a general perspective of how the system works. It can be seen cardboard boxes and bags of garbage which are detected as garbage by the algorithm showing as the final result the set of sub-images containing the garbage.

These network architectures were tested due to the fact that they are all included in Keras applications and that they perform well when used too generic picture classification issues.

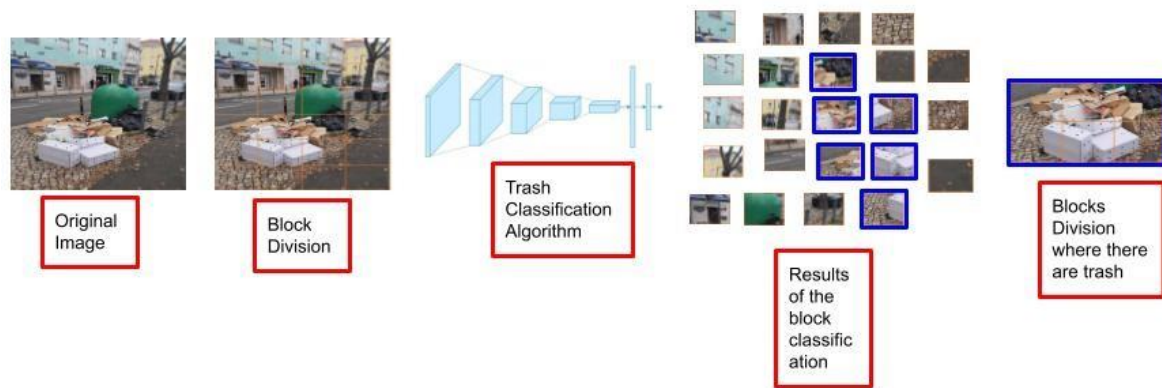


Figure 5. General System Architecture

3.1.2. Object-based garbage detector

To identify the trash outside the containers, YOLO, an object-based garbage detector that uses a real-time object identification algorithm, was utilized. Instead of using the image blocks, the original images were used in which the algorithm identified the area where the garbage is located by assigning a bounding box and label to each identified area, written in an annotation which will be explained in detail later this chapter. Figure 6 shows a sketch of how the image detection system works and Figure 7 shows us an example given by the garbage detection system.

Two types of deep learning models were used for this project. Both, as already mentioned, uses CNN' but the patch-based garbage detection system identifies whether there is garbage in the sub-image. The other uses the YOLO algorithm to detect where the garbage is in the image by giving the percentage of confidence and respective class, which in this case will only be “garbage” since the main focus is to identify where the trash is located.

3.2. Dataset

The process of creating the dataset is covered in this part, from gathering the data through managing it and using it as an input to train the algorithm.

As previously stated, Lisbon City Hall's provided the dataset that was utilised. The process from image acquisition to datasets creation is shown in Figure 8.

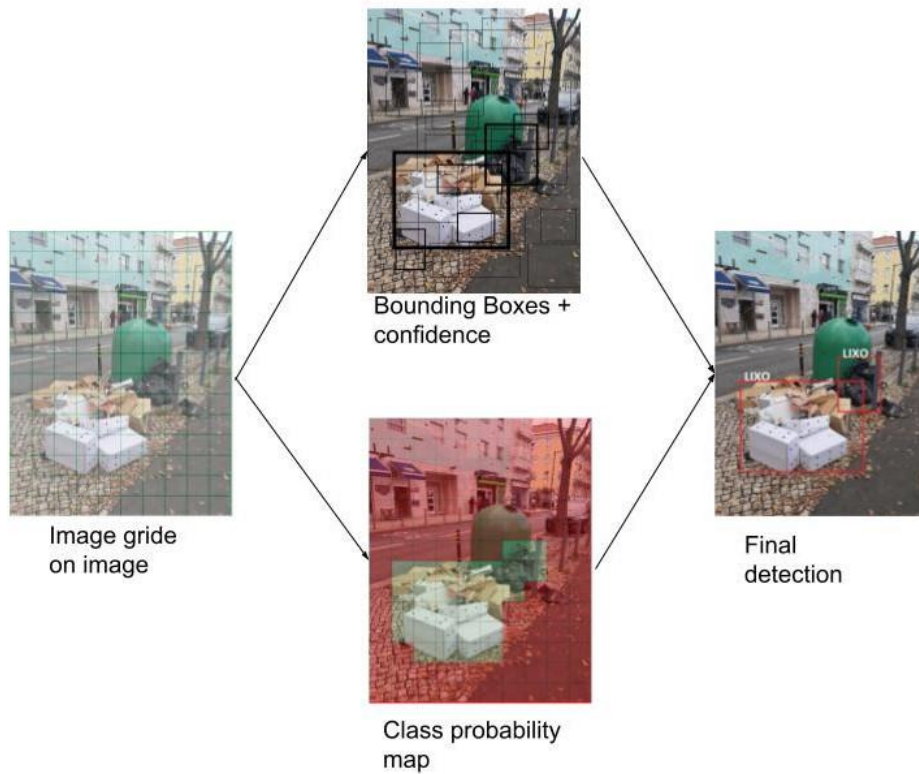


Figure 6. *Garbage detection system behavior*

The process starts with the gathering of images of trash outside the equipment by waste collection inspectors or common residents. The “LxDataLab” image database is the result of merging the images acquired by these different sources. To classify the images, the Labelling program is used to create annotations for each image in which the x and y coordinates are displayed in a box format where the garbage is located. Finally, the training, testing and validation datasets used for different classification models is built.

The following subsections describe how the data is used in the patch-based garbage detector system and in the object-based garbage detector. Both approaches use the same images and annotations, with some distinctions between them, which will be explained.



Figure 7. Example of YOLO classification

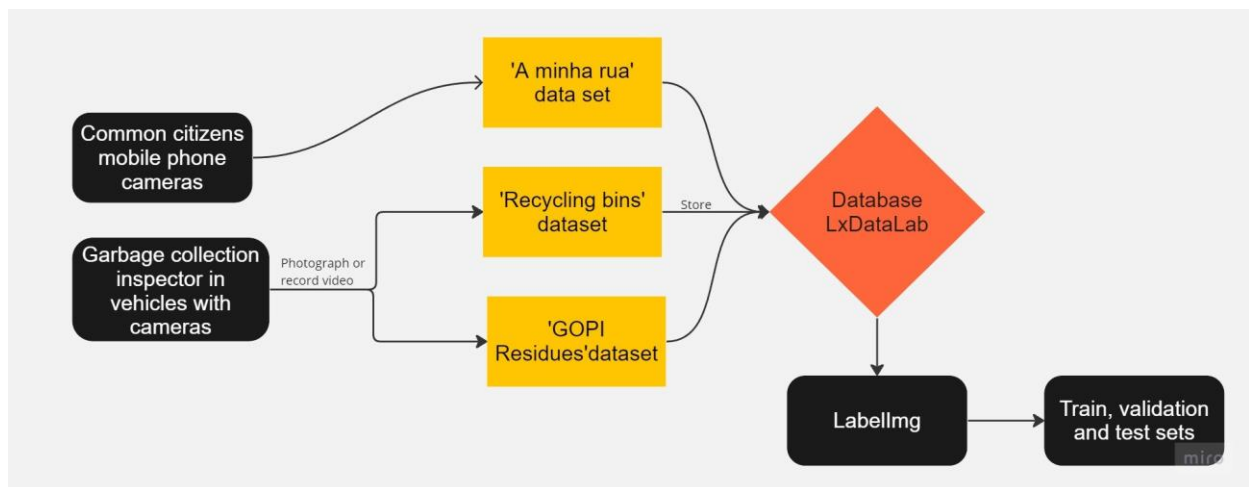


Figure 8. Dataset creation

3.2.1. Data processing and classes

As previously said, the development of the garbage detection system based on supervised learning is dependent on the input data, specifically images. A private archive named “LxDataLab”, run by the Lisbon Center for Urban Management and Intelligence,

receives data from Lisbon City Hall. This repository contains information on several issues where task automation by machine learning may be possible. An assortment of images were gathered from various sources for the misplaced garbage detection challenge. One of these sources is the anonymous user-generated app “A minha Rua”, which contains photos, taken by the common citizen. Example of those images are shown in Figure 9.



Figure 9. “A minha rua” app images

Figure 10 shows images acquired by the garbage inspectors. They basically show the same problem of excessive garbage around the containers.

The LxDataLab team created an unlabeled dataset using images similar to those depicted in the examples. Those images were the ones used in the context context of this Dissertation. A total of 1451 images were available: 1032 from collection garbage inspectors, 259 are coming from 5 videos acquired from moving vehicles in Lisbon and 160 came from “A minha rua” app.

The next step was to create annotations for each images that contain trash. For that task, the Labelling software was used. This software, as shown in Figure 11, was used to manually delimit the garbage location found in the images, thus creating a .xml file



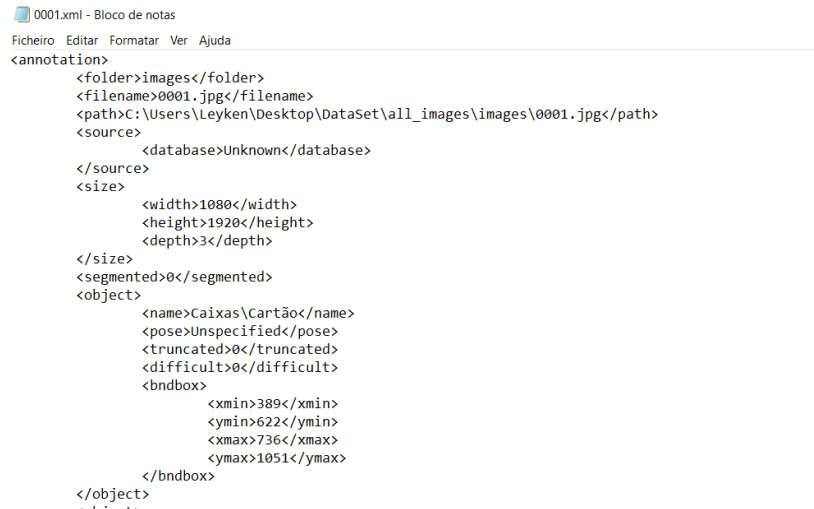
Figure 10. Images taken by Collection Garbage Inspectors

with the coordinates of two points that correspond to a rectangle where the garbage is found. Figure 12 shows an example of an .xml file which was made for the patch-based garbage detector system.



Figure 11. Labelling software

Initially, only 305 images were annotated and it is important to mention that there were images that the program could not open due to their size. Therefore, with this done was created a folder that the respective .xml files with the same names of the images to which they correspond. The file also shows the type of garbage or class it



```

0001.xml - Bloco de notas
Ficheiro Editar Formatar Ver Ajuda
<annotation>
  <folder>images</folder>
  <filename>0001.jpg</filename>
  <path>C:\Users\Leyken\Desktop\DataSet\all_images\images\0001.jpg</path>
  <source>
    <database>Unknown</database>
  </source>
  <size>
    <width>1080</width>
    <height>1920</height>
    <depth>3</depth>
  </size>
  <segmented>0</segmented>
  <object>
    <name>Caixas\Cartão</name>
    <pose>Unspecified</pose>
    <truncated>0</truncated>
    <difficult>0</difficult>
    <bndbox>
      <xmin>389</xmin>
      <ymin>622</ymin>
      <xmax>736</xmax>
      <ymax>1051</ymax>
    </bndbox>
  </object>
</annotation>

```

Figure 12. Example of an .xml file with the bounding boxes coordinates

contains. Initially, the annotation were divided it into 6 classes, “bags”, “boxes/card”, “branches”, “glass”, “plastic” and “undifferentiated” which is garbage that doesn’t have a specific category, for example, mattresses or appliances. However, a change was made using just a single class called “garbage” because there were some classes that contained too few examples and furthermore the focus is on identifying garbage outside the containers.

3.2.2. Specific dataset processing for the object detection-based garbage detector

Regarding the dataset used in the object detection-based garbage detector, an additional processing step was done in order to use it in the YOLO object detection network. The original 305 images in the dataset and the 6 garbage classes were used. However, the format of the annotations created by labellmg is not accepted by the yolo software. For this reason, with the appropriate code, it is possible to convert the labellmg annotations so that they correspond to the standards of the yolo software. Next, annotations containing only the “garbage” class were used in order to focus only on garbage detection. Additionally, more images were annotated, resulting in a total of 428 images available for training and testing the object detection-based garbage detector system. The resulting image data set was split into 80% for training, 10% for validation and 10% for testing.

The annotation accepted by yolov5 is made up of 5 different numbers. The annotation file would look like '0 0.2 0.52 0.34 0.77'. The first number will always be an integer that corresponds to an associated class. For example, the number 3 corresponds to the card class, the number zero corresponds to the bag class and so on. In the case where only one class was used in the annotations, the class associated with the number zero corresponds to the 'garbage' class. Next, there are 4 numbers between 0 and 1, which have been normalized by the dimensions of the image. The first two numbers correspond to the x and y coordinates of the center of the bounding box and the last two numbers correspond to the height and width. Figure 13 helps to understand how it works.

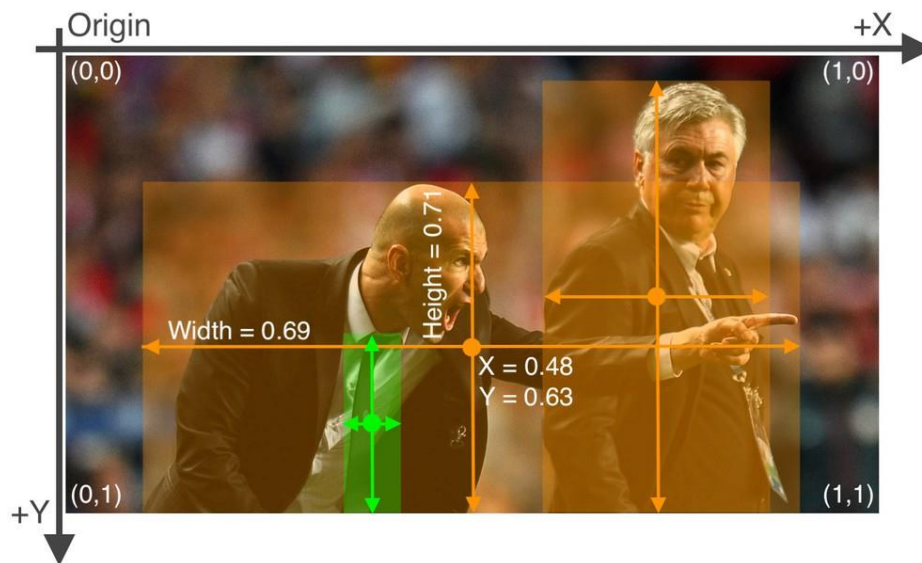


Figure 13. Yolov5 annotations explained ⁷

3.2.3. Image resizing for the patch-based garbage detector system

To facilitate the work of the patch-based garbage detector system, the size of all the images was normalized. However, the aspect ratio of the images varied significantly, including situations portrait and landscape variations. Following this train of thought, it is important to take into account the aspect ratio that the image has when its size is changed to avoid the image contents being “stretched” or “shrunk”. These values depend on whether the image is in landscape or portrait. Therefore, a default image

⁷<https://blog.paperspace.com/train-yolov5-custom-data/>

height value was defined for the portrait images, and a default image width value was defined for landscape images. These values were set to 640 pixels and 460 pixels, respectively. Assuming it is a portrait image, the height value is changed to the default value set, as shown in the equation below. Next, the Factor value would be calculated so that the image is normalized. This number is calculated by dividing the image height by the default value. Finally, the image length value is multiplied by the value obtained in the factor, ensuring that the image's aspect ratio is kept, so that the image looks normal. Mathematical speaking, the following formulas help understand how the values are obtained:

$$\begin{aligned}
 ImageHeight &= DefaultHeight \\
 Factor &= Height / DefaultHeight \\
 NewLength &= Factor * Length
 \end{aligned}
 \tag{3.1}$$

Figure 14 shows a simple visual example of the performed computation and results.

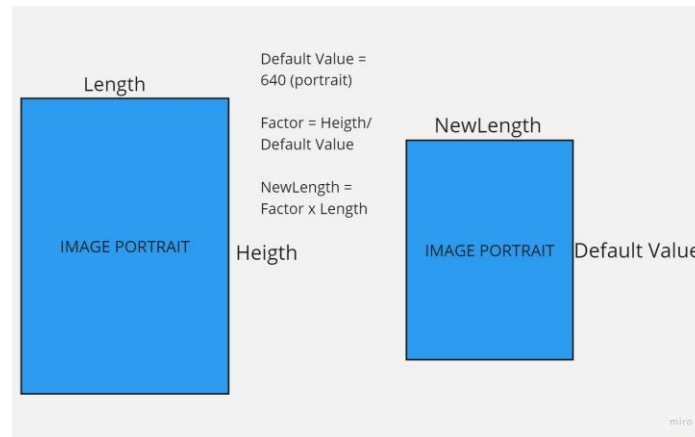


Figure 14. Factor value calculation

3.2.4. Annotation's adjustment for the patch-based garbage detector system

Since the images' size is changed, it was also necessary to change the coordinates of the bounding boxes in the annotations. To change the coordinates a similar reasoning and calculations were used, but instead of taking the values of height and length, the

values (xmin,ymin) and (xmax,ymax) were taken, and a variable named AspectRatio was created. Figure 15 illustrates the calculations.



Figure 15. New Bounding boxes values

The new image annotations also contain the coordinates of the already changed bounding boxes, but this time, as mentioned before, they do not contain a class, just the coordinates, which in itself indicates that it is part of the class “with_garbage”.

```

0021.xml - Bloco de notas
Ficheiro Editar Formatar Ver Ajuda
<annotation>
  <filename>0021.jpg</filename>
  <object>
    <name>0021.jpg</name>
    <bndbox>
      <xmin>82</xmin>
      <xmax>167</xmax>
      <ymin>293</ymin>
      <ymax>385</ymax>
    </bndbox>
  </object>
  <object>
    <name>0021.jpg</name>
    <bndbox>
      <xmin>224</xmin>
      <xmax>298</xmax>
      <ymin>270</ymin>
      <ymax>345</ymax>
    </bndbox>
  </object>
</annotation>

```

Figure 16. New Annotation

Next, the images are cropped into 32 by 32 pixel sub-images. The overlap between predicted bounding boxes and ground truth boxes is measured by Intersection over Union (IoU) method, with scores ranging from 0 to 1. To help determine and split the dataset, the IoU method will be used to classify the sub-image as “with_garbage” or as “without_garbage”. This is applied in the patch-based garbage detector system and where the

sub-image is partially contained in the respective bounding box. In more detail, if the sub-image coordinates corresponding to the original image were contained in the bounding box with a value greater than 50%, the sub-image contains garbage, otherwise it does not. The IoU method helps determine this value in order to make a complete assessment of whether or not the sub-image corresponds to an area that contains garbage.

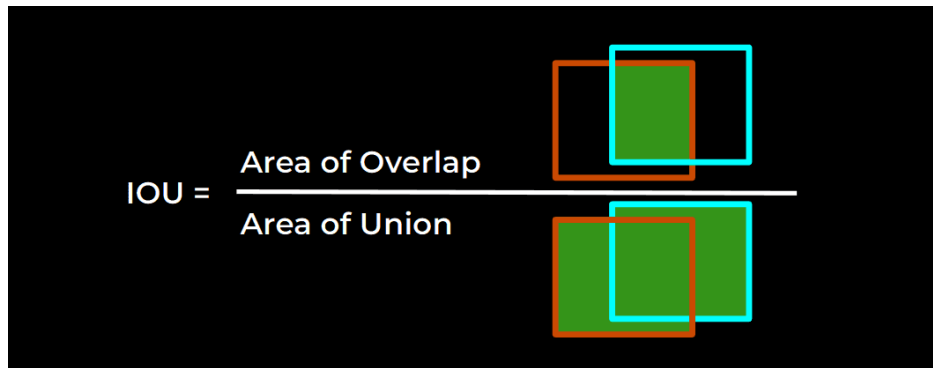


Figure 17. IoU⁸

In the patch-based garbage detector system, the first thing checked was if any part of the sub-image is inside the bounding box using the image coordinates. For example, if in an image the bounding goes from (45,75) to (90,125) and the sub-image is from (0,0) to (32,32) it is easy to understand that this sub-image does not contain garbage. If the overlap area divided by the area of union is greater than 0.5(50%) then it is considered that the sub-image contains garbage. Figure 18 explains in more detail.

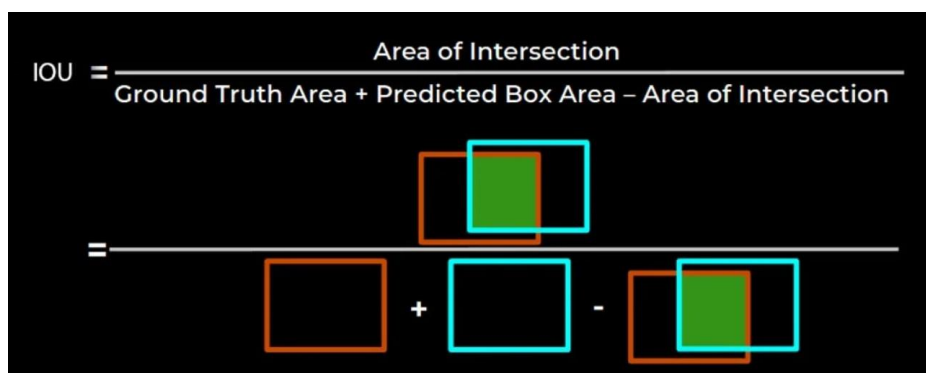


Figure 18. IoU Further Demonstration⁹

⁸<https://learnopencv.com/intersection-over-union-iou-in-object-detection-and-segmentation/>

⁹<https://learnopencv.com/intersection-over-union-iou-in-object-detection-and-segmentation/>

The ground truth area corresponds to the bounding box from the annotations coordinates, and the predicted Box Area, in this case, corresponds to the sub-image which is a 32 by 32 pixel image. When the yolo network was used, the predicted box area references to the prediction made by the algorithm.

Applying this process to the original 305 full images in the dataset, a total of 51,250 sub images were generated: 12,464 images with garbage and 38,786 images without garbage. Initially a 60%/10%/30% training/test/validation split was applied, resulting in 30,749 for training, 15,375 for validating and the remaining 5,125 images for testing. Other splitting percentages were made to verify new results as well. It is also worth mentioning that the “with_garbage” and “without_garbage” classes are imbalanced, with about 3/4 of the samples belonging to the “no garbage” class.

CHAPTER 4

Experimental results

This chapter presents the results for the different experiments carried out for both approaches described in the previous chapter. In addition, the general setup, the programs, the programming language and the installed packages that were used are also described. Afterwards, the chapter presents the results for the patch-based garbage detector system approach, tested with different architectures and configurations. Next, it presents the results and configurations for two different models used in the implementation of the object-based garbage detection system approach. Finally, a general comparison between both approaches is provided.

4.1. General experimental setup

The tensorflow/keras package and the Python programming language were used to create the machine learning models. This package offers source code and enables quick code generation for ML models. The code created in the scope of this dissertation was written in Python and ran on top of Tensorflow. Visual Studio Code was used for the code development and Google Colab was used for training and testing models that followed the object detection-base approach (Yolo-based). Google Colab is a cloud service that is useful for ML and AI research. The python version used was 3.8.13 and the tensorflow/keras API version was 2.4.0. The memory set aside for the object detection-based part of this project was entirely allocated to the computer's CPU because the memory set aside by Google Colab was only temporary. As a result, the network training process took a longer time.

In short, for the patch-based garbage detector system, the python language was used in the Microsoft Studio program, along with the TensorFlow packages mentioned in the previous paragraph. For the object-based garbage detector, Google Colab was used (also using the python language).

All the experiments were performed in a Huawei MateBookD, with a AMD Ryzen 5 2500U CPU, Radeon Vega Mobile Gfx 2.00 GHz GPU, and 8GB of RAM.

4.2. Patch-based approach

Models based on this approach use the dataset containing the sub-images (patches), which were stored into two folders named “with_garbage” and “without_garbage”, according to their class. Two custom CNN were implemented from scratch, trained and tested. In addition, three pre-trained CNN were also experimented using this approach.

For all the CNN architectures used, both the custom models and the pre-trained models, two types of dataset splitting were done: one with 70% for training, 20% for validating and 10% for testing; and the other with 60% for training, 30% for validating and 10% for testing. The settings were the same for both splits and architectures. The use of data augmentation and class balancing using undersampling was experimented on all architectures. However, class balancing using SMOTE was only used on the pre-trained models.

The same data augmentation methods were used in the experiments for each CNN architecture:

- rotation-range = 15
- width shift range = 0,1
- height shift range = 0.1
- shear range = 0.1
- horizontal flip = true
- vertical flip = true

The batch-size value used was always 16 and the image (patch) resolution was 32x32.

4.2.1. Simple CNN from scratch

4.2.1.1. Class balancing using undersampling

Undersampling is a class balancing technique that consists of setting the number of images on each class to the number of samples in the minority class (in this case the one with the fewest patch images). For instance, if the class “without_garbage” has 1000 images and the class “with_garbage” contains 3000 images, 2000 images are removed

from the class “with_garbage” in order to get the same number of images on both classes.

4.2.1.2. Simple 2 convolutional layers CNN

For this CNN architecture, the following configuration were used:

- Batch size of 16
- Image resolution of 32 by 32
- Shuffled order of the images
- Loss function - cross entropy
- Optimized algorithm - ADAM

The architecture was composed of two convolutional layers, one with 16 filters and the other with 32. Each convolutional layer was followed by a Dropout layer, to prevent overfitting, and a MaxPooling layer for subsampling. Afterwards, it uses a Flatten layer and two Dense layers with 128 and 2 neurons respectively. A “relu” activation function was used in the first Dense layer and a “softmax” activation function for the output layer. Initially the use of 100 epochs was tested to see the results and then the same was done using early stopping. The results with early stopping ended around the 7th epoch as figure 19 and figure 20 shows. For this reason the number of epochs was reduced to 20, in order to save time since the efficiency was the same. This way, 20 epochs were used during the training process on all methods.

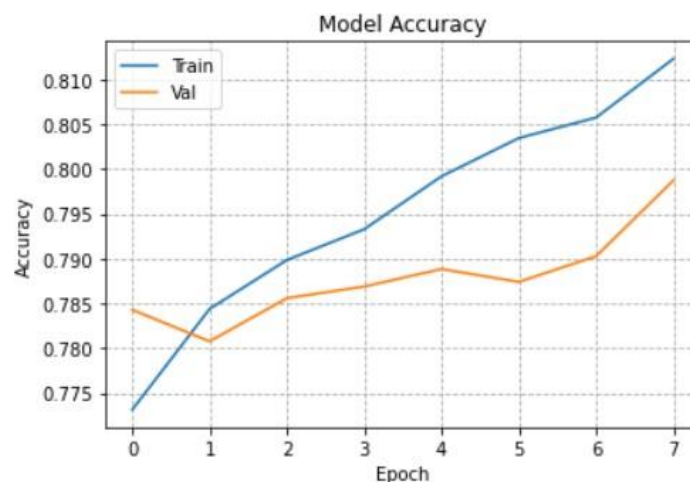


Figure 19. Accuracy plot using Data Augmentation

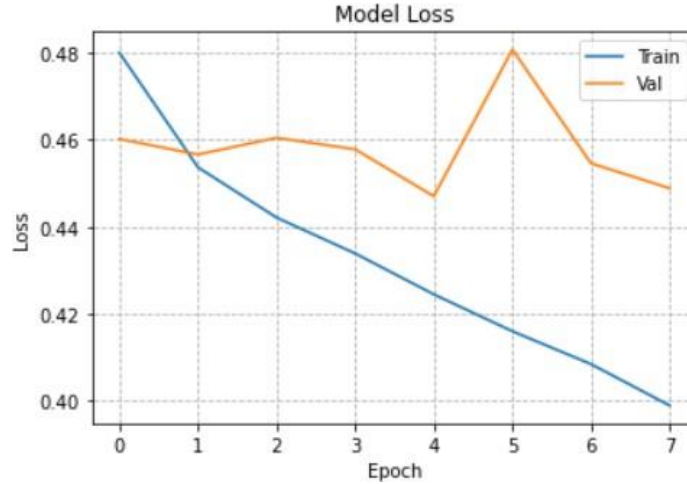


Figure 20. Loss plot using Data Augmentation

2 convolution layers CNN	Train		Validation		Test	
	Acc. [%]	Loss	Acc. [%]	Loss	Acc. [%]	Loss
Without Data Augmentantion	86.92	0.3059	77.83	0.5064	77.15	0.4984
With Data Augmentation	80.47	0.3121	75.99	0.5252	76.42	0.5565
UnderSampling	72.65	0.5429	70.97	0.5429	65.2097	0.6365

Table 1. Simple 2 convolutional layers CNN: results for 60/10/30 dataset split

2 convolution layers CNN	Train		Validation		Test	
	Acc. [%]	Loss	Acc. [%]	Loss	Acc. [%]	Loss
Without Data Augmentation	86.06	0.2935	75.74	0.5325	76.3371	0.5088
With Data Augmentation	81.20	0.3001	76.41	0.5125	76.91	0.4992
UnderSampling	73.93	0.5237	63.14	0.6932	70.0903	0.5497

Table 2. Simple 2 convolutional layers CNN: results for 70/10/20 dataset split

The configuration that showed the best results for both dataset splits was the one that used all images and didn't use data augmentation. Nevertheless, the test results were not satisfactory. Other configurations showed even worse results, not very promising and with significantly worse accuracy values. Using the undersampling method showed worse results, reducing the number of images in order to balance the dataset worsened the accuracy and loss values in the system.

4.2.1.3. Simple 4 convolutional layers CNN

The composition of this network is very similar to the one previously described. However, two additional convolution layers were added, with 64 and 128 filters respectively,

after the 32-filter convolution layer. The number of neurons in the first Dense layer also increased from 128 to 256. The network was also trained along 20 epochs and the same configurations used for the simple 2-convolutional layer CNN were used.

4 convolution layers CNN	Train		Validation		Test	
	Acc. [%]	Loss	Acc. [%]	Loss	Acc. [%]	Loss
Without Data Augmentantion	85.43	0.3124	76.29	0.5344	75.88	0.5081
With Data Augmentation	77.41	0.4683	74.09	0.5287	77.1902	0.4990
UnderSampling	74.01	0.5195	61.12	0.6112	58.7763	0.7421

Table 3. Simple 4 convolutional layers CNN: results for 60/10/30 dataset split

4 convolution layers CNN	Train		Validation		Test	
	Acc. [%]	Loss	Acc. [%]	Loss	Acc. [%]	Loss
Without Data Augmentantion	81.99	0.3332	75.39	0.5444	74.33	0.5121
With Data Augmentation	77.23	0.4738	75.66	0.5053	76.9551	0.4930
UnderSampling	72.69	0.5414	58.92	0.7050	65.0883	0.6392

Table 4. Simple 4 convolutional layers CNN: results for 70/10/20 dataset split

Again, the best accuracy results were achieved when no Data Augmentation was applied. In general, the achieved results were similar, but slightly lower, to the previous case. The gradients used to update the weights during training become smaller and smaller as they propagate across the layers as more layers are added. As a result, the weights of the early layers may not be updated efficiently, causing the model to perform badly.

4.2.2. Pre-trained networks using Transfer Learning

The use of transfer learning required specific data preprocessing procedures for each pre-trained model. Additionally, a different topology in final layers of the network was implemented: a Flatten layer, a 256 neuron Dense layer, a Dropout layer and a 2 neuron Dense layer at the output classification.

For training, the same configurations used in the simple 2 convolutional layer CNN were used.

Because there were insufficient examples of the minority class, imbalanced classification has the disadvantage that a model cannot efficiently learn the decision boundary.

The minority class's examples can be oversampled as one approach to mitigate this issue. Simple replication of samples from the minority class in the training dataset before model fitting can do this. Although it can balance the class distribution, this doesn't give the model any new data¹⁰.

4.2.2.1. Synthetic Minority Oversampling Technique - SMOTE

This technique is basically the opposite of undersampling. Instead of reducing the number of images of the majority class, the number of images of the minority class is increase. However, SMOTE is a statistical method for evenly expanding the number of examples in your dataset. The component creates new instances from the minority class that actually specify as input that already exist. This method was only applied on the pre-trained networks because it promises better results and code wise, with the scratch networks it presented numerous errors making it difficult to show solid results.

4.2.2.2. MobileNet

MobileNets are built on a simplified design that creates lightweight deep neural networks using depth-wise separable convolutions [19]. Two simple global hyper-parameters that successfully balance latency and accuracy are described. These hyper-parameters give the model builder the ability to choose the right model size for their application based on the limits of the problem.

In this case the pre-trained model network trained for 20 epochs.

MobileNet	Train		Validation		Test	
	Acc. [%]	Loss	Acc. [%]	Loss	Acc. [%]	Loss
Without Data Augmentantion	75.89	0.5474	75.74	0.5325	76.3317	0.5088
With Data Augmentation	75.69	0.5356	75.57	0.5437	75.6943	0.5393
UnderSampling	59.14	0.6669	58.58	0.6701	50.4112	0.7064
SMOTE	95.16	0.1504	78.24	0.6656	83.9	0.466

Table 5. MobileNet results 60/ 10/ 30

For this case, the best results are shown by SMOTE not only in precision values but also in loss. SMOTE contributes to more accurate predictions and higher model performance by minimizing bias and capturing crucial properties of the minority class. In

¹⁰<https://machinelearningmastery.com/smote-oversampling-for-imbalanced-classification/>

MobileNet	Train		Validation		Test	
	Acc. [%]	Loss	Acc. [%]	Loss	Acc. [%]	Loss
Without Data Augmentantion	85.53	0.5494	79.43	0.5502	75.9804	0.6357
With Data Augmentation	75.69	0.5389	75.69	0.5447	75.6878	0.5346
UnderSampling	60.99	0.6321	59.78	0.6887	52.0012	0.6892
SMOTE	95.531	0.1492	81.74	0.5197	84.1	0.472

Table 6. MoibileNet results 70/10/20

addition, models trained without data augmentation still give slightly better results than the others.

4.2.2.3. ResNet50

Since accuracy tends to decrease as the number of layers in the neural network increases after a certain point due to the vanishing gradient problem, the Resnet architectures use residual blocks (or "skip connections") to address a problem typically associated with deeper networks. These residual blocks display quick connections that do identity mapping [20]. Only one residual neural network architecture from this family was tested, the ResNet50 model.

The configuration for training the network was the same as the MobileNet case.

20 training epochs were used because early stopping was tested before to see if the use of 100 epochs were unnecessary, which it was. Besides that, the pre-processing function associated was changed to the pre-trained network.

ResNet50	Train		Validation		Test	
	Acc. [%]	Loss	Acc. [%]	Loss	Acc. [%]	Loss
Without Data Augmentantion	75.68	0.5490	75.68	0.5474	75.6813	0.5477
With Data Augmentation	75.68	0.5538	75.68	0.5548	75.6813	0.5517
UnderSampling	70.82	0.6762	62.28	0.6531	69.8232	0.7142
SMOTE	93.78	0.1225	79.04	0.6947	87.2	0.40

Table 7. ResNet50 results 60/10/30

For Resnet50 the accuracy results using SMOTE are also the best, as in the MobileNet case. However, the overfitting problem seem to be worse due to higher loss values in the validation set. The results achieved for both dataset splits are similar both in accuracy and in loss values.

4.2.2.4. DenseNet

ResNet50	Train		Validation		Test	
	Acc. [%]	Loss	Acc. [%]	Loss	Acc. [%]	Loss
Without Data Augmentantion	75.68	0.5494	75.68	0.5453	75.68	0.5483
With Data Augmentation	75.68	0.5453	75.68	0.5525	75.69	0.5438
UnderSampling	71.41	0.6702	63.41	0.6394	70.9232	0.7328
SMOTE	94.75	0.1824	80.42	0.6201	83.9	0.45

Table 8. ResNet50 results 70/10/20

DenseNet builds dense interlayer connections using dense blocks. Every layer transfers its own features to every layer above it while taking extra information from every layer below it. As for the architecture of this network, nothing was added and 20 training epochs were also used.

DenseNet	Train		Validation		Test	
	Acc. [%]	Loss	Acc. [%]	Loss	Acc. [%]	Loss
Without Data Augmentantion	89.74	0.4288	77.55	0.4798	77.1122	0.4703
With Data Augmentation	78.66	0.4537	75.57	0.5437	75.4537	0.4702
UnderSampling	73.93	0.6159	68.15	0.6159	72.77	0.5160
SMOTE	97.69	0.0926	78.50	0.7940	84.1	0.44

Table 9. DenseNet results 60/10/30

DenseNet	Train		Validation		Test	
	Acc. [%]	Loss	Acc. [%]	Loss	Acc. [%]	Loss
Without Data Augmentantion	80.49	0.4132	79.04	0.4631	79.04	0.4631
With Data Augmentation	78.36	0.4693	78.28	0.4764	76.8	0.4762
UnderSampling	72.98	0.5859	69.45	0.5959	73.02	0.5042
SMOTE	97.62	0.089	81.33	0.6580	86.2	0.39

Table 10. DenseNet results 70/10/20

For the DenseNet model, the first architecture showed better results in training precision. This model is advantageous for solving the vanishing-gradient problem, improve feature propagation, promote feature reuse, and significantly reduce the number of parameters. Overall, it showed lower loss values except for validation in the SMOTE settings where it showed worse but a better test percentage.

Yolo Model	Epochs	Number of classes	Epochs	Images used	Batch size	Run
Yolov5s model	100	6	100	305	16	1
Yolov5s model	100	6	100	305	16	2
Yolov5s model	50	6	50	305	16	3
Yolov5s model	50	1	50	305	16	4
Yolov5s model	50	1	50	428	16	5
Yolov5s model	50	1	50	428	16	6
Yolov5s model	50	1	50	428	16	7
Yolov5m model	50	1	35	428	8	8

Table 11. Object detection-based experiences settings

4.3. Object detection-based approach

This is the part that distinguishes from S. Fernandes work in [1], since the previous experiments are based on her idea of dividing images into small images of 64 by 64 pixels, and for this thesis it was divided by 32 by 32 pixels.

The yolov5s and yolov5m models were the two models used to carry out the experiments, which will also reveal their configurations. For yolov5s, one of the datasets used was the original 305 images without any alterations, with two different types of annotations, those containing the 6 classes and those containing only the "garbage" class. Next, the dataset used was the one containing 428 images but only the "garbage" class was used for this part of the experiments as it promised better results.

The following results and graphs will be the experiments performed using the yolov5s model, when adding more images to dataset it will be used other model to test new results. To organize it better, table 11 shows how the runs were organized and their differences. As for the dataset split, 80% was for training, 10% for test and validation for all the runs. The number of workers used for both models was 24. Workers specifies the maximum number of data loaders. This was not shown in the table since it was always the same number.

4.3.1. Object detection performance assessment

Various different types of results were analysed. First, there is precision which corresponds to the number of true positives divided by the total number of positive predictions. A true positive is when a model correctly predicts the predicted class. Whereas

a false positive is when the model incorrectly assigns the wrong class compared to the predicted class. In the context of this dissertation, precision is number of BBs (bounding boxes) detected that actually contained garbage divided by the total number of BBs detected. It is a critical component that determines the accuracy and reliability of an experience's outcomes. Mathematically speaking, to obtain the precision values, the formula is as follows:

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive} \quad (4.1)$$

Analyzing graphs such as figure 21, it is possible to see the evolution of precision values over the epochs.

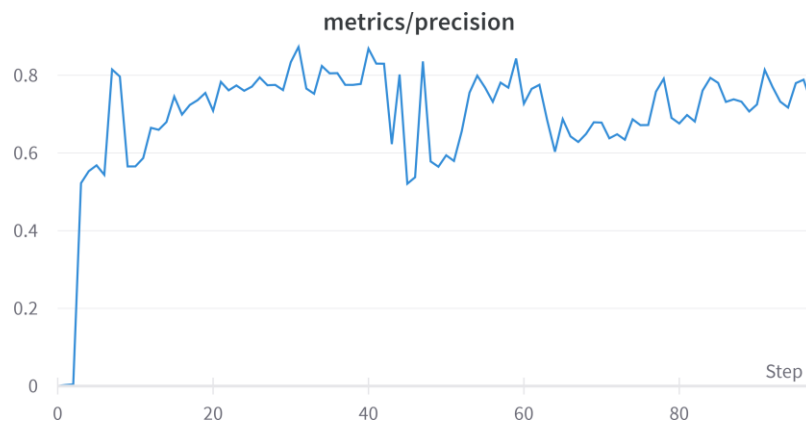


Figure 21. Precision values example

Second, there are the loss values. Analyzing the loss values is important as it also demonstrates whether the algorithm has done a good job by checking that the loss values are low, otherwise it may have overfitting problems that cause the system to lose efficiency. Three types of loss values were obtained:

- Cls_loss stands for loss of object category loss, which calculates the chances that there is an object in the region of interest. It measures the classification error of predicted labels;

- Box_loss indicates how effectively the model can predict the object's location, it denotes the loss of whether it contains the item. It represents the rate to which the detected bounding box fills the labeled one;
- Obj_loss shows how accurately the algorithm can pinpoint an objects center and how completely the anticipated bounding box incorporates an object. A loss metric that determines how "tight" the predicted bounding boxes are to the ground truth objects and is based on a specific loss function.

Finally, there is the precision versus recall curve. Precision and recall are two values which together are used to evaluate the performance of classification or information retrieval systems. A perfect classifier has precision and recall both equal to 1. The explanation of precision has already been given in this chapter, as for recall, it revers to the number of BBs detected that actually contained garbage divided by the total number of BBs where there was garbage. In order to obtain the recall values, the formula is as follows:

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative} \quad (4.2)$$

Before analyzing an image of the Precision vs Recall curve, it is necessary to explain what it represents. Figure 22 helps to understand that the area under the curve is the important factor to consider. The more area it covers, the better the results will be. Analyzing a precision vs recall curve, is especially effective in cases where the number of negatives is significantly more than the number of positives.

4.3.2. Results Evaluation

In this section, the results will be revealed and analyzed. Table 12 shows the precision values and the precision curve values as well.

As it can be seen from table 12, the 6° run obtained the best accuracy value. This run corresponds to the dataset with 428 images, 50 epochs and batch size 16. However, the run that used the yolov5m model obtained a better precision curve value and a solid

¹¹<https://deepchecks.com/f1-score-accuracy-roc-auc-and-pr-auc-metrics-for-models/>

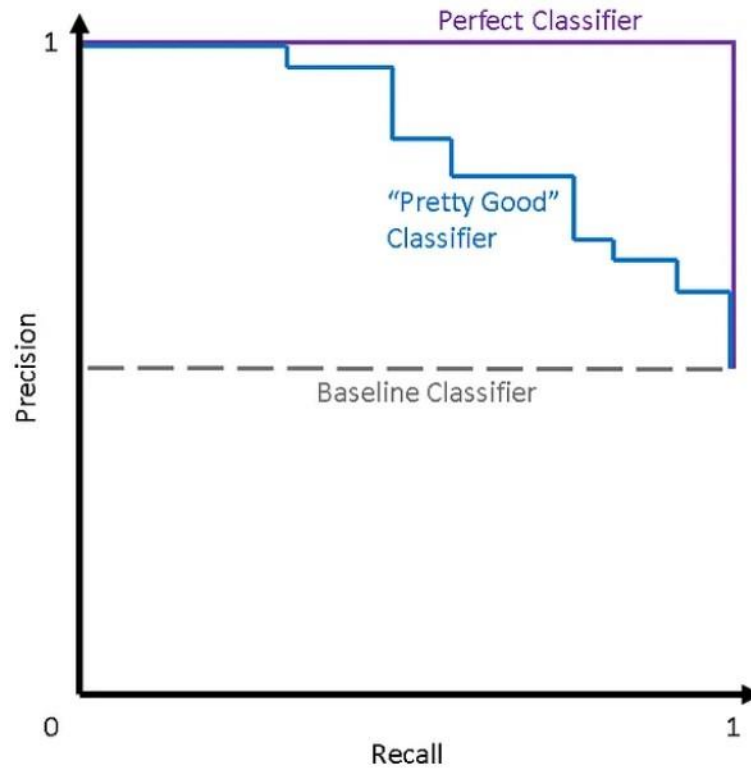


Figure 22. Precision vs recall curve explained¹¹

Run Number	Maximum precision reach	Epoch	Precision Curve Value
1°run	86.9%	9	44.3%
2°run	84.9%	28	41.3%
3°run	85.1%	14	39.3%
4°run	85.6%	23	72.4%
5°run	85.6%	27	79.5%
6°run	88.2%	34	78.4%
7°run	83.8%	39	75.4%
8°run	84.1%	38	79.9%

Table 12. Runs precision results

overall precision value. Looking at 23 where the 88.18% value was obtained, the recall value was 48.7%.

Overall, using more images and only one class made the results more consistent, showing better classification quality Figure 24 shows examples of test results for the 6th run.

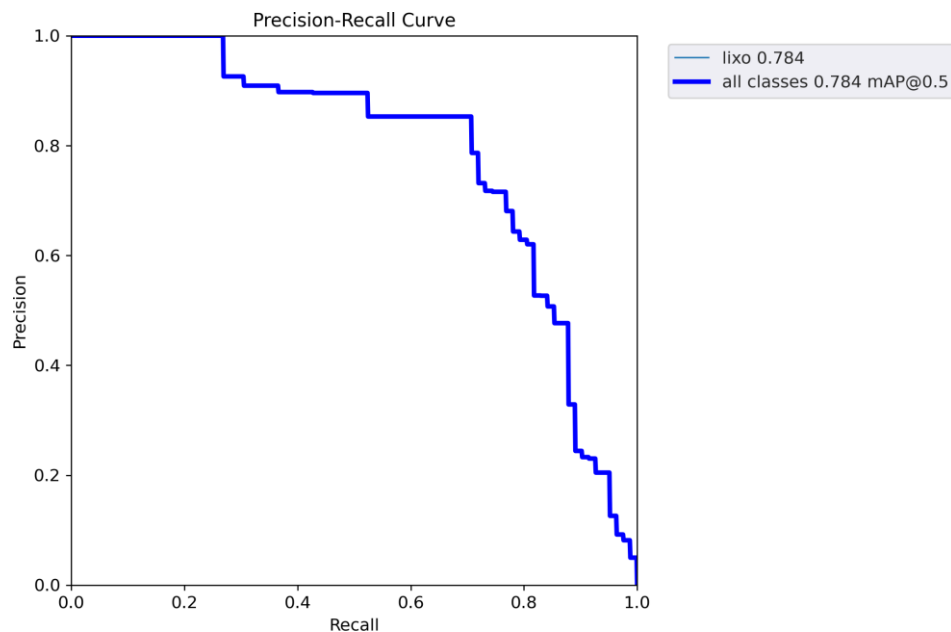


Figure 23. Precision curve run 6



Figure 24. Examples for the test results of the 6^o run

However there are a few results under the 0.5 which are considered false positives. Figure 25 shows a few of those cases. Analyzing the image, there are examples where

the algorithm sometimes has difficulty identifying the garbage that is “deepest” in the image. In addition, the image also reveals two false garbage identifications. The algorithm sometimes finds structures or shapes that could be mistaken for garbage, but the algorithm doesn’t show much confidence in these evaluations.



Figure 25. 4 examples for the test results lower than 0.5

Loss values are divided into train and validation, the lower the value the better. However, for all the runs, the values were basically the same. For the train/box_loss the value are between 0.04 and 0.12 and 0.0275 to 0.0150 for the train/obj_loss and a train/cls_loss stay constantly in 0. As for the val/box_loss the decrease from 0.10 to 0.04 and for the val/obj_loss the values are from 0.0275 to 0.0150. For last, the val/cls_loss also stay in zero.

4.3.3. Patch-based vs. object detection-based results

In this section, the performance of both systems is compared. For doing this comparison, in the patch-based garbage detector system, it was used the altered images and

System	Overall Percentage
patch-based garbage detector system	81.6%
object-based garbage detector system YOLOv5s model	82.2%
object-based garbage detector system YOLOv5m model	82.9%

Table 13. Both systems overall results

colored everything that wasn't garbage black. Then the bounding box with the annotation coordinates corresponding to the area where the garbage was originally located is the white area. The same was done, with the image blocks. All the image blocks that have been classified as garbage will be colored white and the rest black as it can be seen in figure 26. Next, a comparison is made between the two images in which the number of white pixels in the image with the labeled bounding box is compared with the image containing the sub-images, giving a percentage. For this comparison, the IoU method was also used. After that, the values of all the images are averaged.

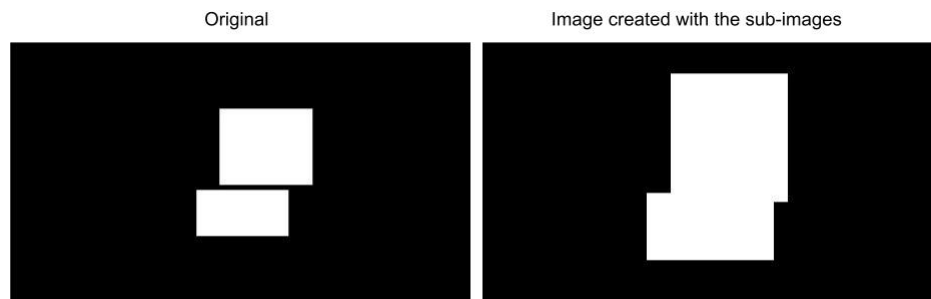


Figure 26. Patch-based garbage detector black and white image distinguish

For the object-based garbage detector, the algorithm allows to get the predicted bounding-box and the ground-truth box that corresponds to the original annotation coordinates. Then using the same IoU method it is possible to get the which percentage of the prediction box is valid.

For the patch-based garbage detector system, 81.6% was obtained and for the object-based garbage detector system for the YOLOv5s model, using the 6° run values, it reached 82.2% and for the YOLOv5m model it was slightly better at 82.9%. Table 13 shows the results.

It should be noted that in terms of accuracy values, the highest value obtained was using the SMOTE method with the pre-trained networks, with values above 90%.

CHAPTER 5

Conclusion

The main aim of this dissertation was to create a system capable of identifying garbage outside the containers using the images provided by the Lisbon City Council. Two systems have been created to deal with this problem.

The patch-based garbage detector was the first system created. As mentioned in section 3.2.2, the dataset images were initially altered to be the same size, making the necessary changes without distorting the images and then changing the new coordinates of the bounding boxes of the annotations. The images were then split into sub-images, which allowed to overcome the small size of the original dataset. The sub-images were labelled into 2 classes named “with_garbage” and “without_garbage”. Finally, the sub-images dataset was split into training, testing and validation sets.

The patch-based garbage detector is a good system for making it easier to identify garbage outside the containers. As presented at the end of chapter 4, using the IoU method mentioned above and placing the “real” box and the box predicted by the system in white and the rest in black in order to obtain an average of the value predicted by the box created by the sub-images helped to visualize and see the potential of this system.

Using the different architectures and adding data augmentation and experimenting with other methods, it was possible to obtain various results for the patch-based garbage detector system. The most promising turned out to be a model that used the SMOTE method for dataset balancing, using ResNet50, DenseNet and MobileNet.

For the object-based garbage detector system, two different groups of classes were used to categorize the garbage. The results of the experiments described in section 4.1.5 showed that using only one class improved the efficiency of the system. Furthermore, increasing the total number of images from 305 to 428 also showed slight improvements

Nowadays, there are better versions of the yolo system, the one used, as indicated several times in this dissertation, was yolov5. However, there is now even yolov8, which promises better performance and results. As the years go by, there will be more and better versions of yolo that promise better results with the same dataset used.

The patch-based garbage detector system contains many images with black garbage bags, and some sub-images were only black with some light reflections and sometimes not even that because the photo was taken at night. These small factors hamper the system's performance

The SMOTE method showed the best results in the patch-based garbage detector system. Possibly, with a larger dataset, not only this method, but also the architecture of the neural network, this same system will certainly show better results. In addition, with a larger dataset it will also be possible to categorize the type of garbage, as this was one of the problems presented. Distinguishing whether it is plastic, glass or cardboard will certainly improve the quality of the system.

With the results obtained, it is possible to answer the research questions in section 1.3. By checking the results obtained and comparing the two systems, it is possible to conclude that the Object detection-based system showed better results by comparing the accuracy of the locations through the average IoU of the areas identified as waste in each of the systems. In the patch-based garbage detector, change all image resolution did not prove to be improve the accuracy results.

With regard to the objectives set out in section 1.4, it was not possible to categorize waste according to the various classes due to a lack of examples, but a new system was explored which promised better results in terms of both accuracy and loss. Furthermore, as already mentioned, with more advanced versions of yolo, it will certainly be possible not only to get better results, but with a greater variety of images in the dataset, it will also be possible to categorize the type of garbage in more detail.

With the aim of making a fair comparison between the two systems, the objective was achieved by using the method explained in section 4.3.3. In this way, a balanced comparison was made between the two systems.

References

- [1] S. H. Fernandes, "Identification of residues deposited outside of the deposition equipment, using video analytics," M.S. thesis, ISCTE-IUL, 2021.
- [2] A. Hevner and S. Chatterjee, *Design Research in Information Systems: Theory and Practice*. Jan. 2010, vol. 22, ISBN: 978-1-4419-5652-1. DOI: 10.1007/978-1-4419-5653-8.
- [3] K. Peffers, P. Tuunanen, M. A. Rothenberger, and S. Chatterjee, "A design science research methodology for information systems research," *Journal of Management Information Systems*, vol. 24, no. 3, pp. 45-77, 2007. DOI: 10.2753/MIS0742-1222240302.
- [4] D. Fox, J. Sillito, and F. Maurer, "Agile methods and user-centered design: How these two methodologies are being successfully integrated in industry," 2008, pp. 63-72, ISBN: 9780769533216. DOI: 10.1109/Agile.2008.78.
- [5] R. Briner and D. Denyer, "Systematic review and evidence synthesis as a practice and scholarship tool," in Jan. 2012, pp. 112-129, ISBN: 9780199763986. DOI: 10.1093/oxfordhb/9780199763986.013.0007.
- [6] A. Liberati, D. Altman, J. Tetzlaff, *et al.*, "The prisma statement for reporting systematic reviews and meta-analyses of studies that evaluate health care interventions: Explanation and elaboration," *Journal of clinical epidemiology*, vol. 62, e1-34, Aug. 2009. DOI: 10.1016/j.jclinepi.2009.06.006.
- [7] M. Rad, A. Kaenel, A. Droux, *et al.*, "A computer vision system to localize and classify wastes on the streets," Oct. 2017, pp. 195-204, ISBN: 978-3-319-68344-7. DOI: 10.1007/978-3-319-68345-4_18.
- [8] R. Stewart and M. Andriluka, "End-to-end people detection in crowded scenes," *Arxiv*, Jun. 2015.

- [9] A. Krizhevsky, I. Sutskever, and G. Hinton, "Imagenet classification with deep convolutional neural networks," *Neural Information Processing Systems*, vol. 25, Jan. 2012. DOI: 10.1145/3065386.
- [10] N. Nnamoko, J. Barrowclough, and J. Procter, "Solid waste image classification using deep convolutional neural network," *Infrastructures*, vol. 7, p. 47, Mar. 2022. DOI: 10.3390/infrastructures7040047.
- [11] J. Cheng and Q. Pang, "Research of waste sorting system based on convolutional neural network," vol. 769, 2021. DOI: 10.1088/1755-1315/769/2/022006.
- [12] M. Valente, H. Silva, J. M. L. P. Caldeira, V. N. G. J. Soares, and P. D. Gaspar, "Computer vision approaches to waste containers detection," in *2019 14th Iberian Conference on Information Systems and Technologies (CISTI)*, 2019, pp. 1-4. DOI: 10.23919/CISTI.2019.8760862.
- [13] —, "Detection of waste containers using computer vision," 2019. DOI: 10.3390/asi2010011. [Online]. Available: www.mdpi.com/journal/asi.
- [14] J. Sousa, A. Rebelo, and J. Cardoso, "Automation of waste sorting with deep learning," 2019, pp. 43-48, ISBN: 9781728153377. DOI: 10.1109/WVC.2019.8876924.
- [15] Melinte, D. Dumitriu, D. Mărgăritescu, and M. Ancuța, "Deep learning computer vision for sorting and size determination of municipal waste," 2020, pp. 142-152. DOI: 10.1007/978-3-030-26991-3_14.
- [16] B. Carolis, F. Ladogana, and N. Macchiarulo, "Yolo trashnet: Garbage detection in video streams," May 2020, pp. 1-7. DOI: 10.1109/EAIS48028.2020.9122693.
- [17] Y. Wang and X. Zhang, "Autonomous garbage detection for intelligent urban management," *MATEC Web of Conferences*, vol. 232, p. 01 056, Jan. 2018. DOI: 10.1051/mateconf/201823201056.
- [18] M. Córdova, A. Pinto, C. Hellevik, *et al.*, "Litter detection with deep learning: A comparative study," *Sensors*, vol. 22, p. 548, Jan. 2022. DOI: 10.3390/s22020548.
- [19] A. G. Howard, M. Zhu, B. Chen, *et al.*, *Mobilenets: Efficient convolutional neural networks for mobile vision applications*, 2017. DOI: 10.48550/ARXIV.1704.04861. [Online]. Available: <https://arxiv.org/abs/1704.04861>.

- [20] M. Shafiq and Z. Gu, “Deep residual learning for image recognition: A survey,” *Applied Sciences*, Sep. 2022. DOI: 10.3390/app12188972.