smart cities

MDPI

Article

Smart City Community Watch—Camera-Based Community Watch for Traffic and Illegal Dumping

Nupur Pathak [1,†], Gangotri Biswal [1,†], Megha Goushal [1,†], Vraj Mistry [1,†], Palak Shah [1,†], Fenglian Li [2,*] and Jerry Gao [3,*]

[1] Department of Applied Data Science, San Jose State University, San Jose, CA 95192, USA; nups.pathak@gmail.com (N.P.); gangotri.biswal@gmail.com (G.B.); meghagoushal@gmail.com (M.G.); mistryvraj3198@gmail.com (V.M.); palakshah.ec@gmail.com (P.S.)
[2] College of Electronic Information Engineering, Taiyuan University of Technology, Taiyuan 030024, China
[3] Department of Computer Engineering, San Jose State University, San Jose, CA 95192, USA
* Correspondence: lifenglian@tyut.edu.cn (F.L.); jerry.gao@sjsu.edu (J.G.)
† These authors contributed equally to this work.

Highlights:

What are the main findings?

- The smart city community watch program developed using YOLOv5 for object detection and DeepSORT for multi-object tracking achieves 97% accuracy in detecting illegal dumping.
- The web-based application integrates person detection, trash detection, license plate detection and extraction, and a decision algorithm aiding government agencies to monitor and effectively manage illegal dumping.

What is the implication of the main finding?

- With the 97% detection accuracy and real-time detection capabilities of the YOLOv5- and DeepSORT-based solution, this solution can help in saving government expenditure to clean up illegal dumping.
- The solution can be integrated with smart city programs such as smart waste management initiatives, aid in effective and proactive public management, and promote public health.

Abstract: The United States is the second-largest waste generator in the world, generating 4.9 pounds (2.2 kg) of Municipal Solid Waste (MSW) per person each day. The excessive amount of waste generated poses serious health and environmental risks, especially because of the prevalence of illegal dumping practices, including improper waste disposal in unauthorized areas. To clean up illegal dumping, the government spends approximately USD 600 per ton, which amounts to USD 178 billion per year. Municipalities face a critical challenge to detect and prevent illegal dumping activities. Current techniques to detect illegal dumping have limited accuracy in detection and do not support an integrated solution of detecting dumping, identifying the vehicle, and a decision algorithm notifying the municipalities in real-time. To tackle this issue, an innovative solution has been developed, utilizing a You Only Look Once (YOLO) detector YOLOv5 for detecting humans, vehicles, license plates, and trash. The solution incorporates DeepSORT for effective identification of illegal dumping by analyzing the distance between a human and the trash's bounding box. It achieved an accuracy of 97% in dumping detection after training on real-time examples and the COCO dataset covering both daytime and nighttime scenarios. This combination of YOLOv5, DeepSORT, and the decision module demonstrates robust capabilities in detecting dumping. The objective of this web-based application is to minimize the adverse effects on the environment and public health. By leveraging advanced object detection and tracking techniques, along with a user-friendly web application, it aims to promote a cleaner, healthier environment for everyone by reducing improper waste disposal.

Keywords: smart city; illegal dumping; video surveillance; computer vision; scene classification; deep learning; big data

check for updates

Citation: Pathak, N.; Biswal, G.; Goushal, M.; Mistry, V.; Shah, P.; Li, F.; Gao, J. Smart City Community Watch—Camera-Based Community Watch for Traffic and Illegal Dumping. *Smart Cities* **2024**, *7*, 2232–2257. https://doi.org/10.3390/smartcities7040088

Academic Editor: Pierluigi Siano

Received: 27 June 2024
Revised: 29 July 2024
Accepted: 2 August 2024
Published: 7 August 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

The practice of illegal dumping involves the disposal of trash or litter on the sides of the road or at non-designated locations [1]. Neighborhoods are frequently littered with old furniture, building supplies, and garbage. Due to its negative effects on the environment, this has become a major problem. To maintain a safe and clean environment, it is prohibited to discard goods in this manner. To mitigate the health risks associated with illegal dumping, this issue must be addressed. The aim of selecting this problem is to raise public awareness of illegal dumping's environmental and public health [2] hazards, and to minimize waste by stopping the action and punishing those responsible. Every year, about 1.5 million tons of illicit garbage is dumped in the United States, and the situation has worsened during the pandemic. In 2019, 48 percent of people aged between 18 and 29 were more likely than all other age groups of people to engage in intentional illegal dumping [3]. According to California, CalRecycle has an estimated 250,000 illegally stockpiled scrap tires. It is estimated that the illegal dumping of mattresses, appliances, furniture, and other items on the pavement in San Francisco neighborhoods results in a significant burden on taxpayers and community resources to remove these unwanted items [4]. Despite California's focus on public safety, residents' complaints and frustrations over unlawful dumping and traffic offenses have not been adequately addressed by the jurisdiction.

We have developed an integrated model for the detection of illegal dumping based on a mobile application and dashboard that provides real-time alerts regarding illegal dumping. A real-time alerting system to identify unlawful dumping in the city would enable the relevant authorities to act against it. There are three primary objectives of the project: Automatic Number Plate Recognition (ANPR), Object Detection, and Action Detection. The purpose is to identify the perpetrator and type of waste dumping by accurately detecting the features of the vehicle and the trash item. An automated monitoring system identifies illegal dumping by using the timestamp, location, and features of the vehicle and waste.

This paper proposes a solution based on cognitive computing technologies for detecting garbage, particularly bulky waste, in locations where it should not be, and notifying the local municipality. Initially, OpenCV is used to determine the license plate number of the vehicle involved in illegal dumping. The next step involves identifying waste items at various locations, including roadside areas. For object detection, YOLOv5 is used in conjunction with OpenCV. With YOLO [5], multiple objects are detected, classified, and labeled with bounding boxes based on the entire image. Considering the unpredictable nature of waste, the focus is on the types of waste that are commonly discarded. The detection of illegal dumping involves identifying individuals who disembark from vehicles, walk along roads, and dispose of trash in unlicensed locations. A Recurrent Convolutional Neural Network (R-CNN) [6] is used to detect illegal dumping incidents based on video input. With this combination of YOLOv5, DeepSORT, and the decision module, an accuracy of 97% is achieved in identifying trash, demonstrating robust capabilities in detecting illegal dumping.

Structure: This paper is organized as follows. Section 2 summarizes the related work. The preparation of training and testing data is described in detail in Section 3. Section 4 describes the methodology for each selected model and the ensemble models that are proposed. Section 5 presents the results and a case study. In Section 6, the results of this study are discussed in the context of other studies of a similar nature. The conclusion of this paper can be found in Section 6.

2. Related Work

2.1. Literature Survey

Table 1 provides a comparison of the various research papers and their purposes, datasets, approaches, and results regarding license plate recognition, action detection, and object detection. Most of the researchers have adopted Region-based Convolutional Neural Network (R-CNN) [7,8], VGG [7], YOLOv3 [9], and ResNet [2,3,10] to achieve a higher accuracy. The highest accuracy can reach 95.1% [11]. However, illegal dumping is still

being investigated, and little progress has been made. Many papers have been written that detect various activities, but there are not many that detect illegal dumping since it would take more data and a clear explanation of what is being thrown away. The aim of this paper is to propose a model that supports the detection of illegal dumping actions in real-time with a high degree of accuracy.

Table 1. Literature survey of illegal dumping detection methods.

Reference	Region	Purpose	Input Parameters	Model	Results
[3]	USA	Identify timing and location of illegal dumping actions from closed-circuit television (CCTV) feeds	RGB Images (Data: ImageNet dataset, Google Images with equal proportion of w/trash and w/o trash)	ResNet	Accuracy of 60.3%
[4]	South Asia	Detect human actions for ambient assisted living (AAL)	IR Images (Data: 5278 images sampled from thermal videos)	LeNet	Accuracy of 87.4%
[12]	Europe	Detect abandoned objects (AOs) using edge information	Video dataset (Data: PETS2007, AVSS2007, CDNET2014, ABODA)	Stable edge detection, clustering	Precision, recall, accuracy, F-measure
[13]	South Asia	Develop Automatic Number Plate Recognition (ANPR) system	RGB Images (Data: 50 images captured from a digital camera)	Optical character recognition (OCR), Radial Basis Function (RBF), Probabilistic Neural Network (PNN)	Accuracy
[14]	Europe	Identify illegal landfills through scene classification in aerial images	RGB Images (Data: 3000 images provided by the Environmental Protection Agency of the Region of Lombardy (ARPA))	ResNet50 and Feature Pyramid Network (FPN)	Precision of 88%, recall of 87%
[10]	USA	Develop an edge-based smart mobile service system for illegal dumping detection and monitoring	RGB Images (Data: dataset with 9963 images and 24,640 annotated objects provided by the Environment Service Department in San Jose)	R-CNN with VGG, Inception V3	Accuracy of 91.3%
[7]	East Asia	Detect garbage dumping actions in surveillance camera footage	RGB Images (Data: COCO dataset with 330,000 images)	R-CNN, R-PCA, and CNN	Accuracy of 68.1%
[8]	Europe	Develop a cloud-based cognitive computing solution to counteract illegal dumping in smart cities	Video (Data: surveillance videos captured by the municipality and security agencies)	TrashNet Densenet121, DenseNet169, InceptionResnetV2, MobileNet	Accuracy of 95.1%
[9]	All	Garbage detection technique in video streams	RGB Images (Data: Google images—2265)	YOLO v3	Precision of 68%

2.2. Technology and Solution Survey

In Table 2, the models used for the task of detecting license plates, objects, and actions are listed along with their advantages and disadvantages.

Table 2. Comparison of models used in detection.

Model	License Plate Detection	Object Detection	Action Detection	Advantages	Disadvantages
ResNet [3,11,14]		x		1. Able to detect features in images and classifies the output of trash bag class detection. 2. Improved accuracy with transfer learning and data augmentation.	1. Constrained by a small dataset and small positive videos. 2. Naive approach to dumping of the plastic bag. 3. Managing a large video database is difficult.
FPN [14]		x		1. Beneficial in object classification with same class objects having variable sizes. 2. Does not require manual bounding box creation.	1. High computational complexity. 2. Requires a large training dataset.
LeNet [4]	x			1. Easy interpretability and simplicity. 2. Can be trained on a small dataset.	1. Limited capability for intricate representations.
Inception [10]		x		1. Able to learn complex patterns and features and achieve high accuracy. 2. High computational efficiency.	1. Limited real-time capabilities for object tracking due to speed.
R-CNN [8,10]		x	x	1. Faster computation with accuracy selecting the Region of Interest (ROI) in images.	1. Treats each frame independently and is unable to capture temporal information.
3D CNN [15]		x	x	1. Useful in object identification, gesture recognition, human action recognition, and aberrant behavior detection. 2. Emphasizes spatial and temporal characteristic extraction in videos.	1. Restricted approach owing to imbalanced category distribution and the paucity of samples.
YOLO [16,17]	x	x		1. A fast YOLO is better for brief sequences.	1. Trade-off of accuracy levels with speed.

Different deep learning models [18] are utilized in the field of computer vision for different types of tasks. In terms of object detection, ResNet [2,3,10] excels at detecting features and classifying trash bags to an extremely high level. There are many advantages to this method, including transfer learning and data augmentation; however, it has some limitations due to small datasets and lacks sophistication in handling scenarios involving plastic bag dumping. FPNs [14] assist in the detection of objects by dealing with variable-sized objects without the need to manually determine bounding boxes, but their complexity and dataset requirements are high. In license plate detection, LeNet [4] is employed as a result of its simplicity and interpretability, but it faces challenges when representing complex data using intricate representations. While Inception [7] boasts high accuracy and computational efficiency in its ability to detect and track objects in real-time, it falls short when it comes to tracking them in real time. As R-CNN [7,8] focuses on ROIs, computation speed is increased by capturing and analyzing the temporal information of each ROI, reducing its ability to detect actions. Despite the fact that 3D CNN [6] is excellent at tracking objects and actions, emphasizing spatial and temporal characteristics in videos, it is constrained by an imbalanced category distribution and a small sample size. YOLO [15] is used for detecting license plates (LPs) and actions, and offers speed benefits but compromises accuracy as a result. Table 3 details the various methods adopted for illegal dumping detection, where the last one lists the accuracy of our Integrated Illegal Dumping Detection model for the classification of person detection, trash detection, LP detection, and character detection. Apparently, our approach gives the best accuracy of 97%.

Table 3. Comparison of existing methods for illegal dumping detection.

Purpose	Reference	Details	Dataset	Model	Accuracy
CNN for object detection and classification	[19,20]	LP detection and character detection	UFPR-ALPR and SSIG SegPlate	YOLOv4-tiny and modified CR-NET	78%
CNN for object detection	[9]	Garbage detection technique in video streams	Google Images	YOLOv3	68%
CNN for action detection	[7]	Detect garbage dumping actions in surveillance cameras	COCO dataset, self-collected	R-CNN, R-PCA, and CNN	68%
Integrated Illegal Dumping Detection model for classification	Our approach	Person detection, trash detection, LP detection, character detection, and decision algorithm	COCO, TACO, Waymo, UFPR-ALPR, authors collected dataset	YOLOv5, DeepSORT, Tesseract OCR	97%

3. Data Engineering

3.1. Data Collection

To detect illegal dumping, the project is divided into three sub-tasks and a module is developed for each sub-task. These sub-tasks are as follows:

- Object detection;
- License plate detection;
- Action detection.

The datasets are collected separately for each sub-task module and for the final combined model. Image datasets comprise both vehicles and people for the model to be able to recognize vehicles and pedestrians, as shown in Figure 1.



Figure 1. Sample images from datasets: (**a**) COCO; (**b**) TACO; (**c**) Waymo; (**d**) UFPR-ALPR; (**e**) authors collected videos.

Table 4 lists the datasets selected for each sub-task. These datasets are selected based on their image content and suitability for the respective models. The COCO dataset provides a wide range of labeled images, whereas the TACO dataset provides the most comprehensive trash database available on the Internet. The two datasets were used for the sub-tasks of object detection and action detection. For the purpose of detecting, reading, and storing the license plate information of suspected illegally dumped vehicles, a large image dataset consisting of vehicles and their license plates is required, preferably in a variety of lighting and weather conditions. The UFPR-ALPR dataset was selected because it includes both vehicle and license plate images, as well as day and night images. In order for the overall model to perform the combined task, we have collected our own dataset which contains illegal dumping actions and includes object detection and action detection, in addition to license plate detection. In total, 180 videos have been collected using an iPhone 13 Pro Max.

Table 4. List of datasets used by sub-task.

Sub-Task	Dataset	Videos	Description
Object detection	COCO [21]	123,000	COCO dataset has a total 330 K images out of which >200 k images are labeled. It supports 1.5 million object instances spanning 80 object classes.
Action detection	TACO [17,22]	1500	TACO dataset presently contains 1500 images of litter with 4784 annotations and 3746 images.
Action detection	Waymo [23]	1000	Waymo dataset has 1000 images from vehicle cameras during day and nighttime with high-quality labels for 4 object classes.
License plate detection	UFPR-ALPR [24]	450	UFPR-ALPR dataset contains 4500 fully annotated photos (nearly 30,000 LP characters) from 150 cars in real-world circumstances in which both vehicle and camera (inside another vehicle) are moving.

Table 4. *Cont.*

Sub-Task	Dataset	Videos	Description
Combined task	Authors collected videos	180	Video dataset of dumping actions which includes object detection and action detection along with license plate detection.

3.2. Data Pre-Processing

The datasets collected from different sources are stored on an Amazon S3 cloud server. The following procedures are used to clean the raw datasets in video and image formats:

- Images were pre-processed using sharpening and augmentation techniques from the source;
- A random sample of the image data was taken and bounding boxes were displayed to determine the quality of the data. It was possible to identify any images that were of poor quality or did not contain the desired information using this method.

To validate the accuracy of the video data, an action detection module was used to break down sample images and frames containing illegal dumping activities. These images and frames were manually labeled for the throwing action and license plate recognition. The video data were then used to train a model capable of detecting illegal dumping. Each image has the following information available in a text file:

1. The vehicle's type (car or motorcycle);
2. The license plate layout (Brazilian or Mercosur);
3. Text (e.g., ABC-1234);
4. The four-corner position (x, y) for each image.

In addition to the license plate bounding box, the corners have been labeled to facilitate the training of the models that explore license plate rectification and data augmentation. LabelImg, a graphical image annotation tool, is used to label images for object detection. Figure 2 illustrates the images from the UFPR-ALPR dataset with LabelImg annotations for license plate detection.

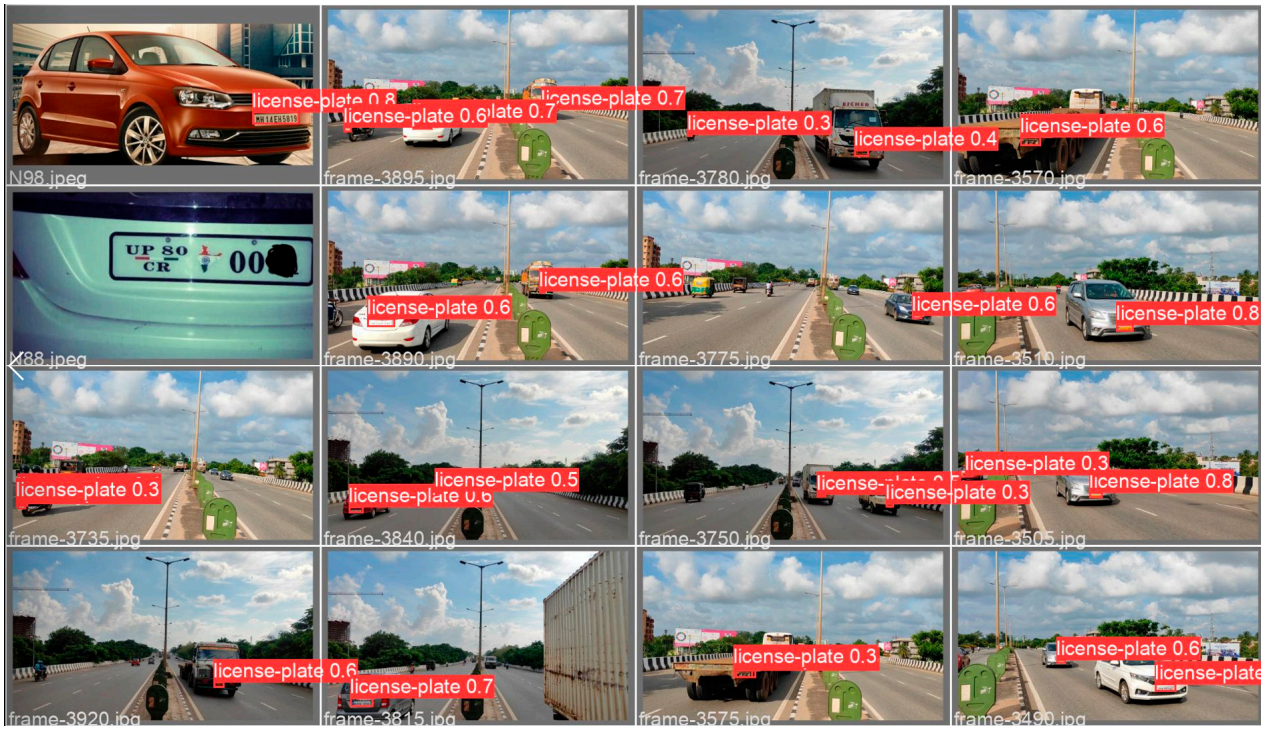


Figure 2. Detecting license plates from UFPR-ALPR dataset and Labellmg.

Figure 3 illustrates an example of an image using LabelImg on a sample image. YOLOv5 is compatible with annotations that correspond to the lines of a text file with the same name as the image.



Figure 3. Sample image using LabelImg.

The authors collected 180 video recordings of 60 s each. Based on an initial review of a few dumping videos, it was decided to re-aggregate video segments into continuous clips in order to capture the entire episode within one video clip. The input video was broken down into images per frame and fed into the model. The raw data consist of 30 frames per second, which were then resampled to 1 frame per second. Further, the frames were interpreted by the model and the final output for model classification was given. These pre-processing videos resulted in 1 min 1 frame per second videos for each of the video clips composed of 60 images per video. The model interprets the frames and produces the final classification result. As illustrated in Figure 4, converting a video into image sequences involves several steps of pre-processing. The following are the modification techniques applied to the images:

- Image resizing (1080 × 1080);
- Grayscale to pre-process the images;
- Gaussian blur and sharpness adjustments;
- Modify the perspective of monitoring cameras at various heights.

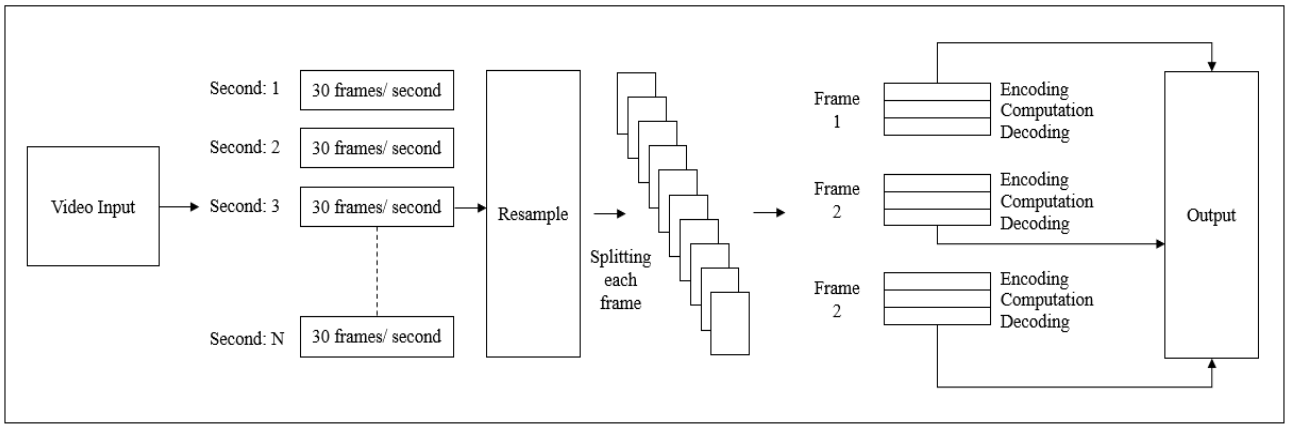


Figure 4. Video pre-processing steps.

The sequence of images is treated as a single unit, with the same alteration applied to each image in the sequence. With resizing, all images are resized to 1080 × 1080 pixels RGBA (R = red, G = green, B = blue, A = alpha for transparency). Most camera systems presently have built-in infrared cameras to improve night vision. Grayscale was used to pre-process the images, as infrared cameras produce grayscale videos. Gaussian blur and sharpness adjustments were used to enhance model performance for low-resolution cameras with resolutions ranging from 240p to 1080p, as showcased in Figure 5. Random perspective modification was used to modify the perspective of monitoring cameras located at various heights capturing targets from a variety of angles. The label formats were converted from YAML to YOLOv5.



(**a**) (**b**) (**c**)

Figure 5. Image transformation techniques as: (**a**) adjusted image sharpness; (**b**) adjusted image color; (**c**) image with Gaussian blur.

3.3. Training Data Preparation

After pre-processing, the data are split into training, validation, and testing. The training dataset is used to train all of the selection models, while the validation dataset is used to compare and validate the models' performance, and the testing dataset is used to determine how general our models are. The division of data follows a distribution of 80% for training, 10% for validation, and 10% for the test set.

4. Model Development

4.1. Model Proposal

There are two modules in the proposed smart city community watch architecture, which are the object detection module and the dumping detection decision module. In the object detection module, there are three sub-modules: person detection, trash detection, and license plate detection and extraction. Figure 6 illustrates the detailed modeling architecture workflow.

The system uses three sub-modules to detect illegal dumping activities: license plate detection, person detection, and trash detection. License plate detection uses YOLOv5 and Tesseract OCR [25] to detect and read license plates. Person detection uses YOLOv5 to detect people involved in illegal dumping activities. Trash detection uses YOLOv5 to detect the type of trash that was dumped. The three sub-modules work together to track

the perpetrator and the trash they are dumping. DeepSORT [26], which is a multi-object tracker, is used to track the movement of objects in the video. It tracks the appearance of the objects, velocity, and motion to be able to identify the person and vehicle that are dumping the trash. The decision module analyzes the data gathered from the three sub-modules in order to determine whether there has been any illegal dumping. When there is a change in activity between two timestamps, such as someone picking up trash and dumping it at another location, the module looks for that change. If the distance between the bounding box of the human and the trash is less than a certain threshold and then increases, then an illegal action in dumping the trash will be detected. The module will alert the relevant authorities if it determines that there has been illegal dumping.

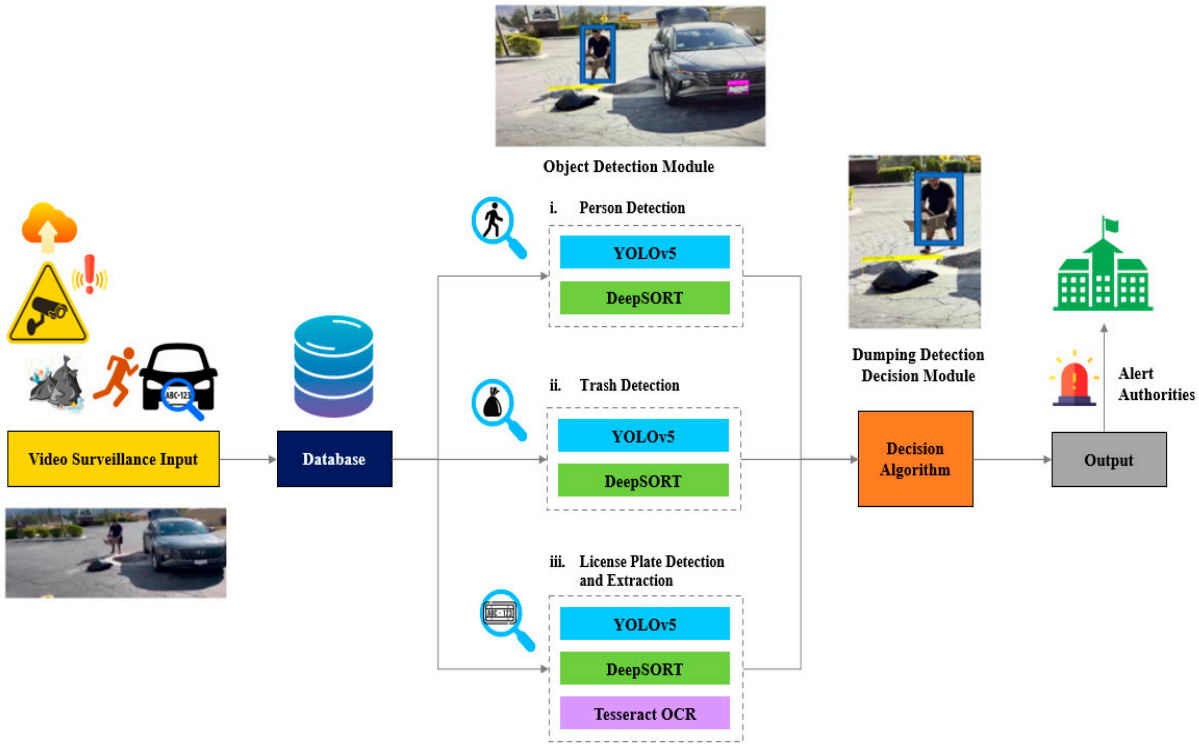


Figure 6. Modeling architecture flow.

YOLOv5 focuses on a module that divides the input data into three parts and processes them differently. The first part is processed with a convolution layer with 32 filters of filter size 3 and a leaky ReLU activation function. The second and third parts are processed with the same convolution layer, but with different dilation rates. This allows YOLOv5 to better handle objects of different sizes. YOLOv5 was trained on the COCO dataset, which contains over 1.5 million images with annotations for 80 object categories. The results of the experiments showed that YOLOv5 achieved a mean average precision (mAP) of 42.3% on the COCO test set. This is a significant improvement over YOLOv4, which achieved a mAP of 39.8%. YOLOv5 is a powerful object detection model that can be used for a variety of tasks. It is fast, accurate, and easy to train.

In the YOLOv5 + DeepSORT mode, the main objective is to detect objects within a video. The Kalman filter plays a crucial role in DeepSORT. In order to track objects, bounding boxes and class probabilities are initially derived from the YOLOv5 model. While continuously error checking, it estimates the current state using the available detections and prior forecasts [27]. The Kalman filter extracts four variables from the YOLO v5 output to determine the center coordinates, aspect ratio, and height of the bounding box. A further challenge is to independently process fresh detections and link these to the most recent predictions of "tracks". This is accomplished through DeepSORT, which combines deep learning with a distance metric (squared Mahalanobis distance). The DeepSORT creators include the appearance feature

vector from a straightforward classifier as the final distance metric since the Kalman filter overlooks several real-world circumstances and camera angles. The velocities of the variables, as well as u′, v′, a′, and h′, are used to forecast the potential course of the item. When an object is no longer visible, the program erases its id. A number of license plate datasets from the U.S. were gathered online in order to train Yolov5 to recognize license plates based on the location of the vehicle. Upon recognizing a license plate, the bounding box coordinates are photographed and recorded as an image. OCR (optical character recognition) [28] is then used to extract the license plate, which is then stored in our database as a string. Figure 7 illustrates the architecture of Yolov5 for object detection and DeepSORT for tracking.

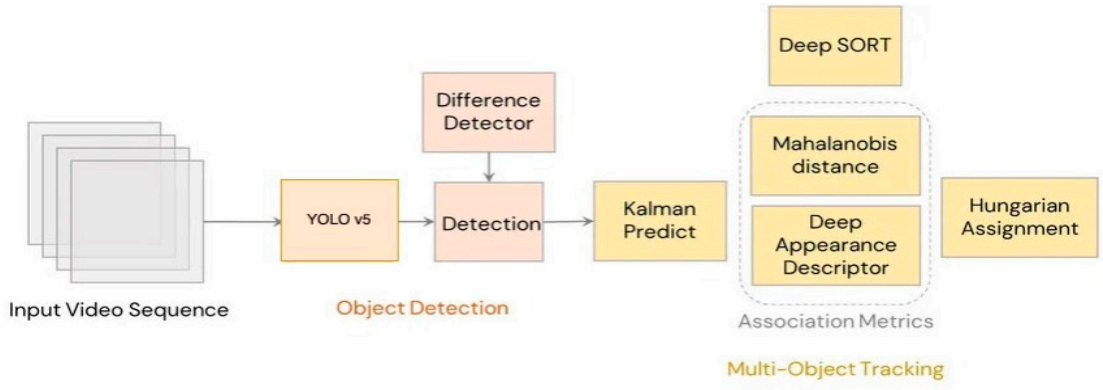


Figure 7. Architecture of YOLOv5 and DeepSORT.

The video output of the YOLOv5 model includes bounding boxes for humans, trash, and license plates. On the basis of Euclidean distance, the motion detects illegal dumping. Video input can be used to determine if a human is holding trash based on the overlap of the bounding box. Furthermore, if a human drops the trash within a reasonable distance, that is considered one step. Another stage occurs when the human bounding box increases in distance from the trash bounding box (in our case, 75+ Euclidean distance is selected as it resulted in achieving relatively higher accuracy scores). If two stages meet their requirements, it is detected as illegal dumping. Based on this approach, Figure 8 illustrates the illegal dumping action solution flowchart.

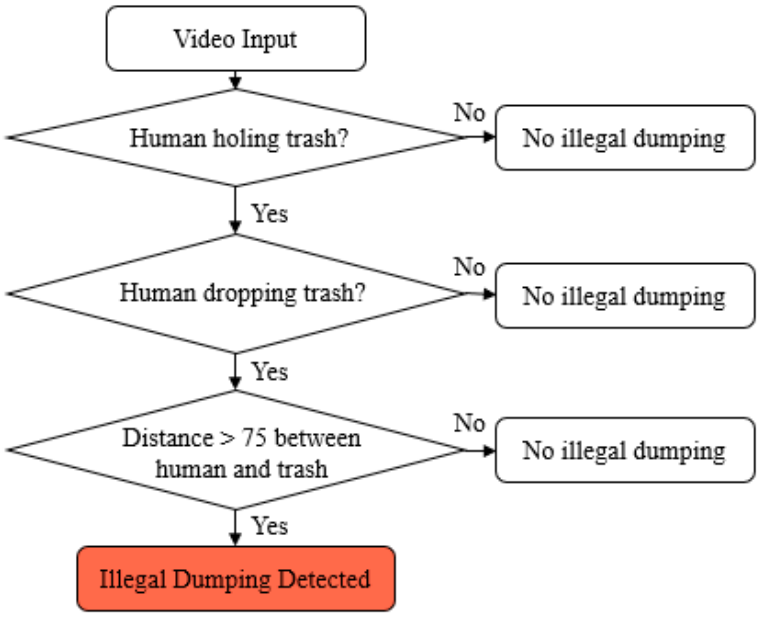


Figure 8. Dumping detection flowchart.

4.2. Model Supports

In this project, NVIDIA Tesla K80 GPU access using Google Colab Pro is used to process large amounts of data along with TensorFlow 2.0. and PyTorch to implement the machine learning models. The data are stored in the local repository and backed up in Google Drive. TensorFlow graph visualization with TensorBoard is used for visualization. Figure 9 shows the system development pipeline. The CCTV cameras capture video from the location. During an illegal dumping activity, the three sub-modules—license plate detection, person detection, and trash detection—are alerted to the location of the perpetrator. License plate detection helps detect the person(s) involved in a timely manner.

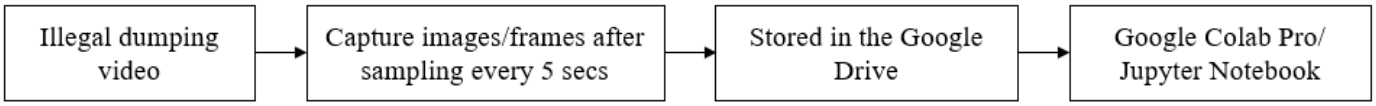


Figure 9. Development pipeline.

All three sub-modules use YOLOv5 for object detection. Tesseract OCR reads license plates. OCR converts 2D images to machine-readable text. Tesseract is an open-source OCR engine which extracts text from images and outputs them to a new text or PDF file. There are a number of stages involved in this process, including the component analysis phase, word and line recognition, and a two-step process to accurately read the word. Tesseract is pretrained, requires minimal setup, and has a high performance with high accuracy. We use open-source tools to pre-process the data to create clear images. YOLOv5 and DeepSORT [29] are used for person and action detection. DeepSORT provides a tracking algorithm based on motion. So, when the perpetrator dumps objects, motions like walking and throwing can be tracked. We then implement a decision module which finds the difference in activities between timestamps to alert the required end-users.

4.3. Model Comparison

Computer vision has become the most commonly used technology in the industry in recent years. Images must be processed in order to examine their details. We discuss and compare the following algorithms for the implementation of image processing algorithms—CNN, R-CNN, YOLO v3, YOLO v4, and YOLOv5 in Table 5. This paper uses YOLOv5 as it has a number of advantages, such as high speed and excellent accuracy for the classification task.

Table 5. Comparison of image processing algorithms.

Model	Advantages	Disadvantages
CNN	Good for classification of objects	Slow and less accurate
Faster R-CNN [30]	Fast and uses RPN	Not suitable for real-time detection
YOLOv3 [31]	Real-time detection	Cannot detect small objects
YOLOv4 [23,32]	High accuracy and speed	Less accuracy and speed than YOLOv5
YOLOv5 [33]	Highest accuracy and inference speed	Higher training time

4.4. Model Evaluation

4.4.1. Performance Evaluation

Mean average precision (mAP) is a measurement used [34] to assess the performance of object detector models like faster R-CNN [35] and YOLOv5. The mAP score is calculated by comparing it to the bounding box which is the ground truth to the original image and the bounding box that is created or detected by the model. The model's detections are more precise the higher the score. Predictions from object detection algorithms consider a class classification label and a bounding box. We gauge the degree to which the anticipated bounding box and the actual or true bounding box overlap for each bounding box. Depending on the numerous detecting problems present, the mAP score is determined by

averaging the average precision (AP) across all classes and/or the overall IoU (Intersection over Union) thresholds. The mAP is calculated by taking the average of the AP for all classes denoted by Q_R.

$$\text{mAP} = \frac{1}{|\,Q_R|} \sum_{q \in Q_R} \text{AP}\,(q) \tag{1}$$

IoU is used to measure the overlap of the bounding boxes. We use the IoU value for a specific IoU threshold to determine precision and recall for jobs involving object detection. Changing the IoU threshold allows us to obtain distinct binary FALSE or TRUE positives for a prediction.

$$\text{Intersection over Union (IoU)} = \frac{\text{Area of Overlap}}{\text{Area of Union}} \tag{2}$$

Precision is used as an evaluation measure.

$$\text{Precision}\ (\%) = \frac{\text{TP}}{\text{TP} + \text{FP}} \times 100 \tag{3}$$

Recall, also known as the positive rate, measures how likely it is that the actual thing is identified.

$$\text{Recall}\ (\%) = \frac{\text{TP}}{\text{TP} + \text{FN}} \times 100 \tag{4}$$

Here, TP expresses true positive, FP expresses false positive, and FN expresses false negative.

4.4.2. License Plate Detection and Recognition Using YOLOv5 and OCR

Character Error Rate (CER) is a key metric used to assess the accuracy of Tesseract-based OCR models. The CER incorporates substitutions (Ss), deletions (Ds), and insertions (Is) into character recognition while being lenient in the presence of minor alignment errors. It provides an intuitive and accurate measure of performance by comparing these errors to the number of characters in a reference text (N), with a CER of 0 indicating perfect recognition. Optimal performance is achieved with a CER score of 0, with 0 being the ideal score. The following notations are used in Equation (5)—S denotes the quantity of substitutions, D denotes the number of detections, I denotes the number of insertions, and N denotes the total number of characters in the reference or ground truth text.

$$\text{CER} = \frac{(\text{S} + \text{D} + \text{I})}{\text{N}} \tag{5}$$

4.4.3. Frames per Second

In addition to the metrics previously discussed, we also consider how fast our model performs, which is also one of the reasons for choosing YOLOv5 as one of our main models so that we can detect and classify illegal dumping actions in real time. We consider how many test frames are processed per second and make that our criterion for achieving this objective.

4.5. Experimental Results

Table 6 shows the evaluation metrics for the models. Medium Yolov5 with about 21.2 million parameters gave the best results for the detection of trash, license plates, and humans. As compared with other models, Yolov5 was better and had a trade-off with the size of images compared to YOLOv3 but achieved greater accuracy in terms of mAP0.5 and the mAP0.5–0.95 confidence interval. As seen in Figure 10, all of the results in the output are shown with more than 70% confidence for its bounding box.

Yolov5 was trained on the GPU with eight workers and a batch size of 16 images of $256 \times 256 \times 3$. All of the images were reshaped before feeding them into the model. The results in Table 6 show the models' performance on them. For fine-tuning the model, Adam

optimizer is used with a 0.01 learning rate and 0.1 with a learning scheduler every five epochs. The model was trained for 150 epochs with pretrained weights with 0.4 label smoothing.

For the vehicle license plate detection, Tesseract OCR was used. The model was pretrained on ALPR dataset weights with more than 18,000 images of vehicle augmented. The model was given a selected region of the output with more than 80% accuracy to extract license plate text. Moreover, before extracting the text, thresholding and sharpening kernels were used to reveal the text even in blurred images. Likewise, the OCR model's performance was evaluated based on the CER to make sure we extract the exact license number for further use. For evaluating the performance of the model, we use weights and a bias interface to check the performance of the model evaluated based on the mAP score.

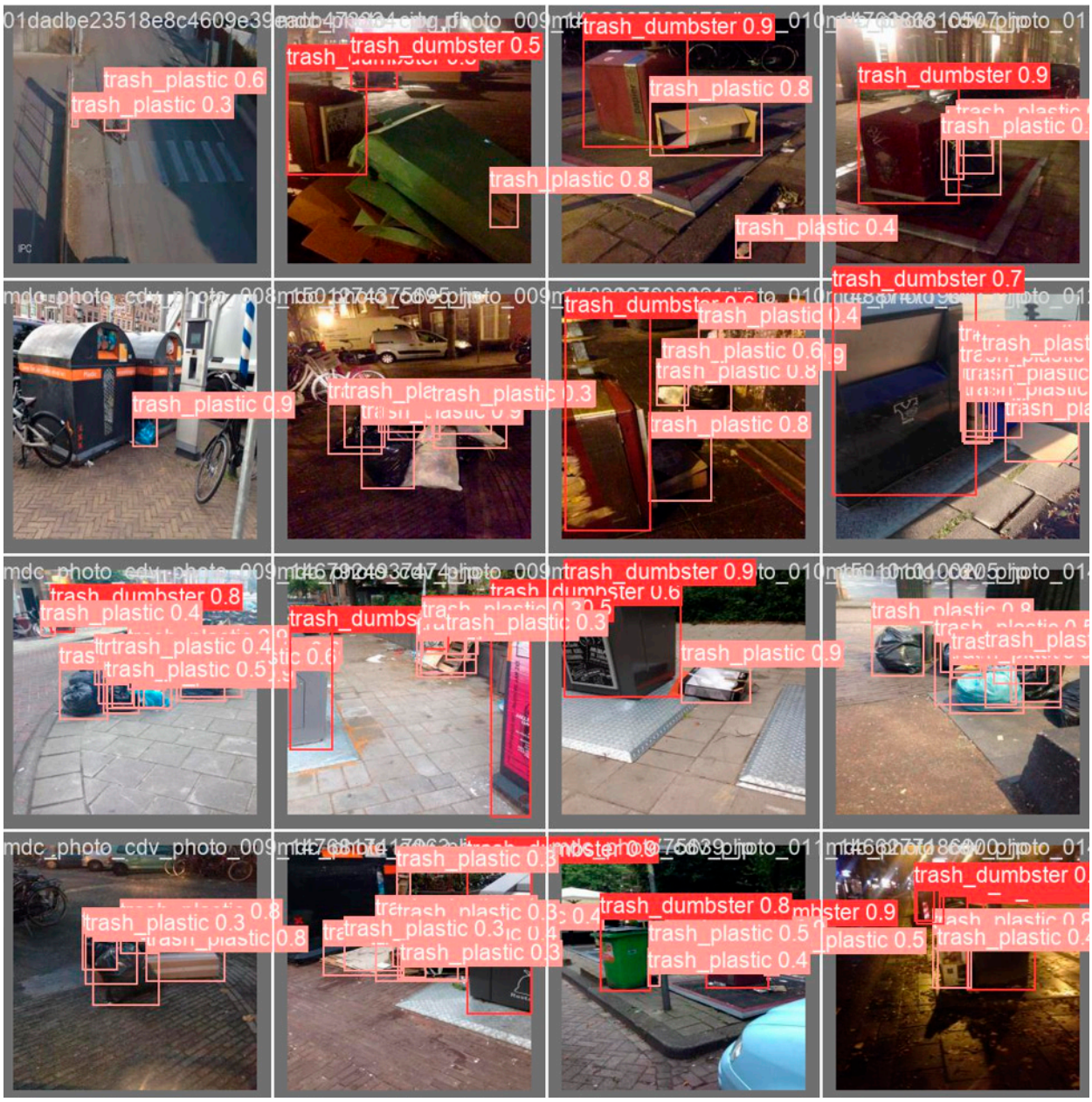
Table 6. Evaluation metrics for different models.

Method	mAP0.50	mAP0.5–0.95	Precision	Recall	Batch Size	Input Resolution
Medium Yolov5	0.74	0.62	0.89	0.78	16	256 × 256
Yolov3-Spp	0.69	0.54	0.81	0.75	16	416 × 416
Faster RCNN	0.67	0.31	0.64	0.48	8	416 × 416



(**a**)

Figure 10. *Cont.*



(**b**)

Figure 10. (**a**) Validation data with actual values; (**b**) predicted boxes by YOLO.v5.

4.5.1. License Plate Detection Model

The performance of the license plate detection model was evaluated using key metrics: mean average precision (mAP), precision, and recall. The model achieved an mAP score of 97.7%, indicating high accuracy in detecting and classifying license plates across the tested scenarios. This high mAP score reflects a well-balanced trade-off between precision and recall, demonstrating the ability of the model to minimize both false positives and false negatives effectively. This relatively high accuracy of metrics indicates that our model has a high prediction performance and we can track license plates using the DeepSORT algorithm. Precision, which measures the accuracy of positive detections, was also notably high, underscoring the model's reliability in identifying correct license plates. The recall metric was strong, highlighting the model's ability to detect nearly all instances of license plates present in the dataset. As illustrated in Figure 11, the performance metrics improved with increasing training epochs, showcasing the model's enhanced capability over time.

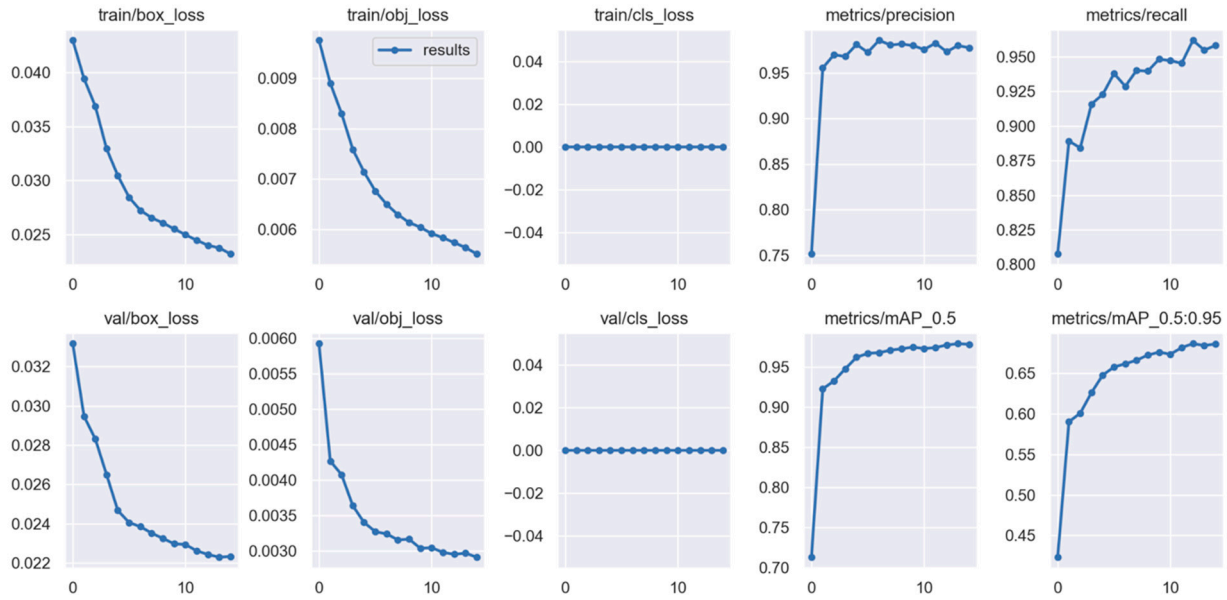


Figure 11. Vehicle license plate model evaluation.

Figure 12 further confirms the model's high classification accuracy, with a successful identification rate of 97% for license plates, validating its effectiveness for real-world applications and integration with tracking algorithms such as DeepSORT.

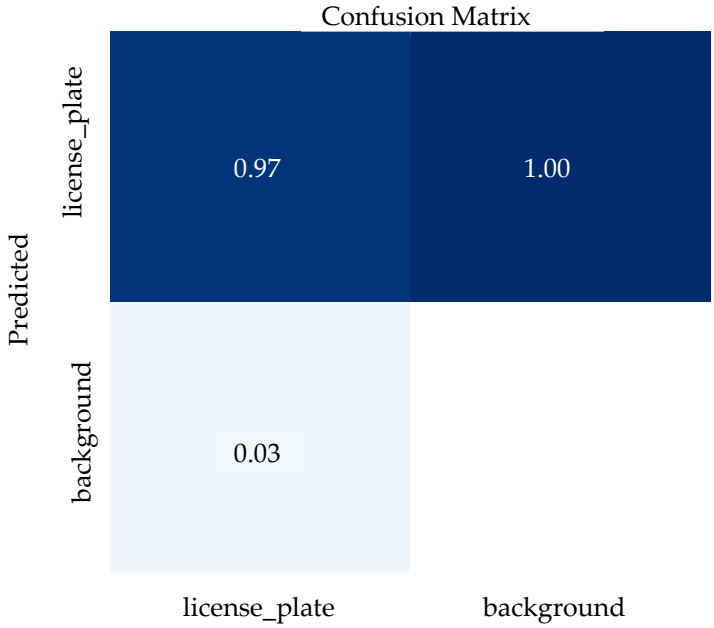


Figure 12. Confusion matrix for vehicle license plate model evaluation.

4.5.2. Trash Detection Model

There are several types of trash detected by the trash detection model, such as cardboard, containers, and garbage bags. A confusion matrix and an mAP score were used during the training process to evaluate the model. Based on the test set, Model 1 with three output classes (containers, garbage bags, and cardboard boxes) classified 79% of containers, 44% of garbage bags, and 34% of cardboard boxes. Model 1 achieved scores for mAP of 0.54 and for recall of 61%. In the event that trash and humans are detected and the distance between these bounding boxes is greater than 75+ Euclidean distances, then the matter is classified as illegal dumping.

A modified version was tested to evaluate the classification task. In the modified model (Model 2), garbage bags and cardboard boxes were combined into a single class. Figure 13 depicts the classes identified by the models and confusion matrix provided in Figure 14.

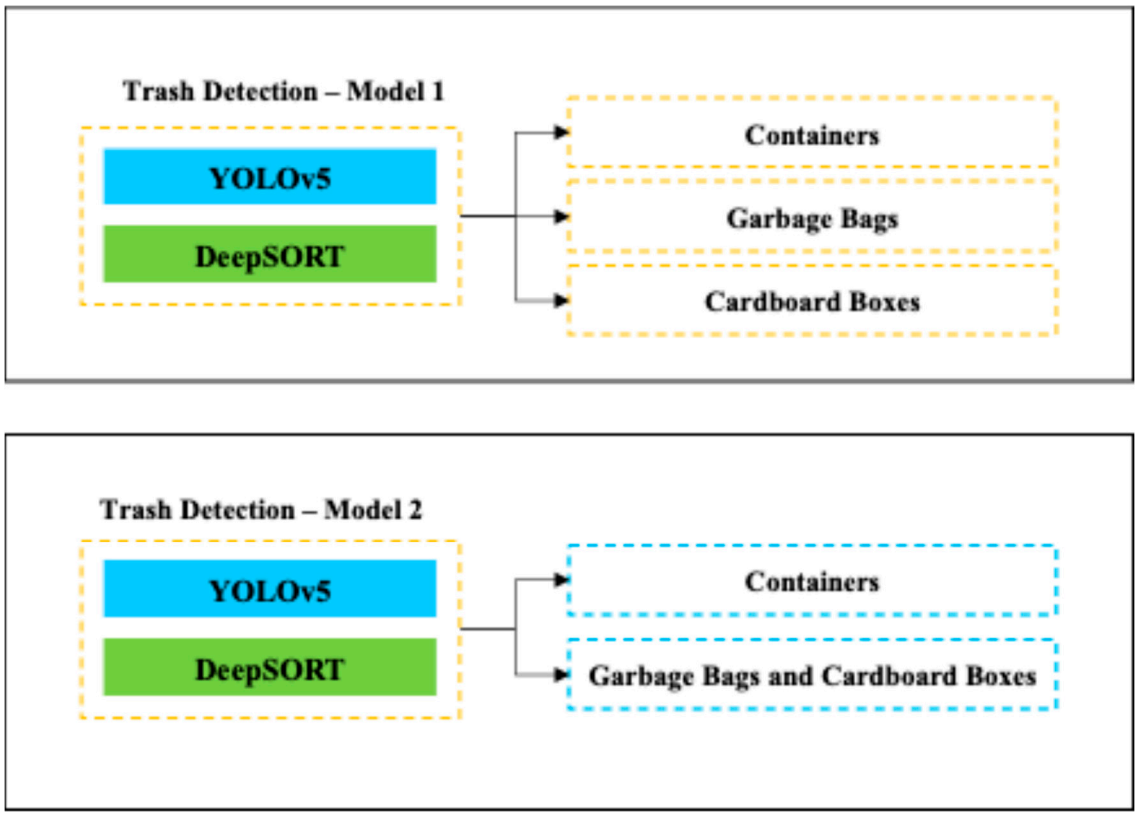


Figure 13. Trash detection by Model 1 and Model 2.

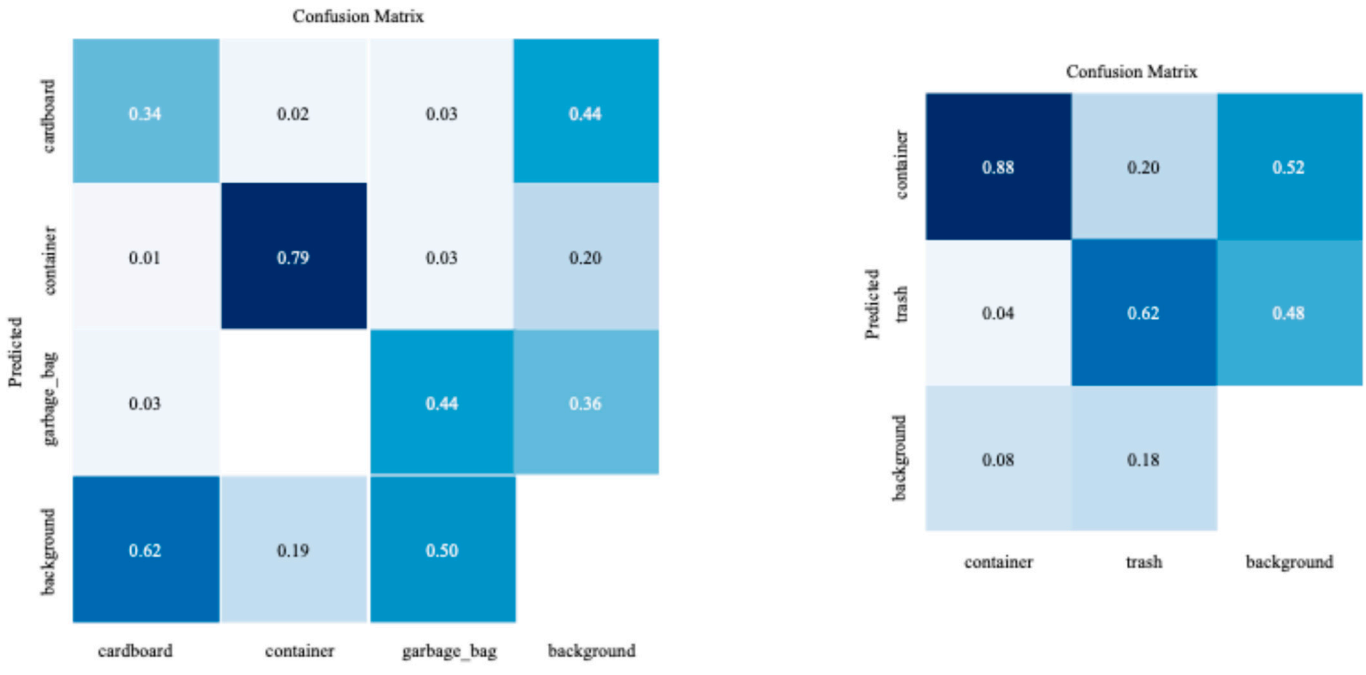


Figure 14. (**a**) Confusion matrix of Model 1; (**b**) confusion matrix of Model 2.

Model 2 achieved an mAP score of approximately 0.74, as depicted in Figure 15 and Table 7. Model 2 correctly classified 88% of containers and 60% of trash, including garbage bags and cardboard boxes. The trash detection model's performance evaluation is illustrated in Figure 16.

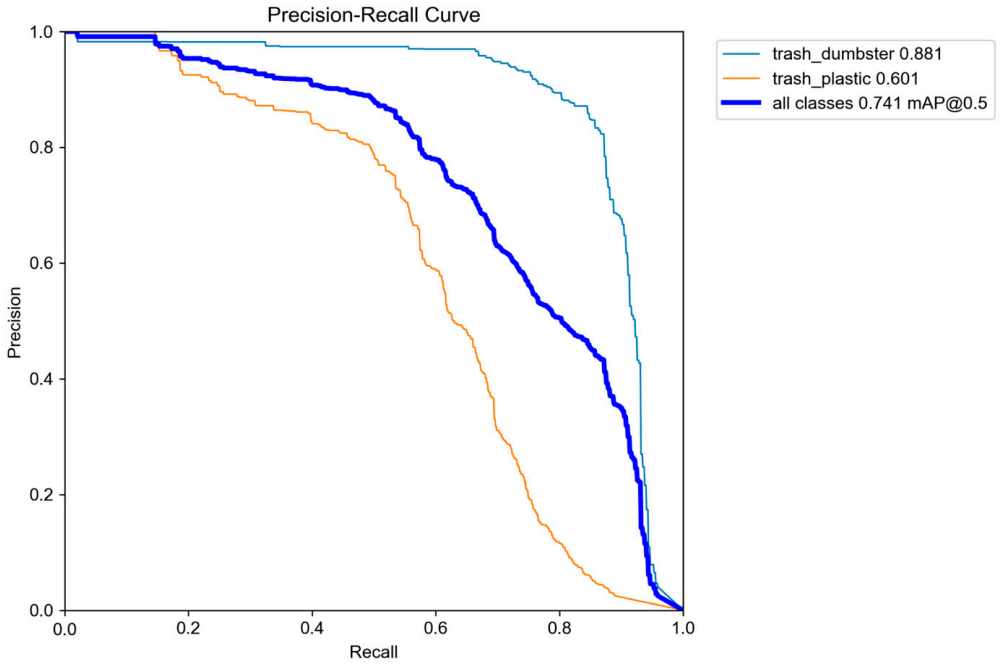


Figure 15. Precision–recall graph for trash detection (Model 2).

Table 7. The performance difference between trash detection Model 1 and Model 2.

Measure	Model 1 (3 Output Classes)	Model 2 (2 Output Classes)
Precision	0.647	0.831
Recall	0.612	0.724
mAP	0.549	0.741

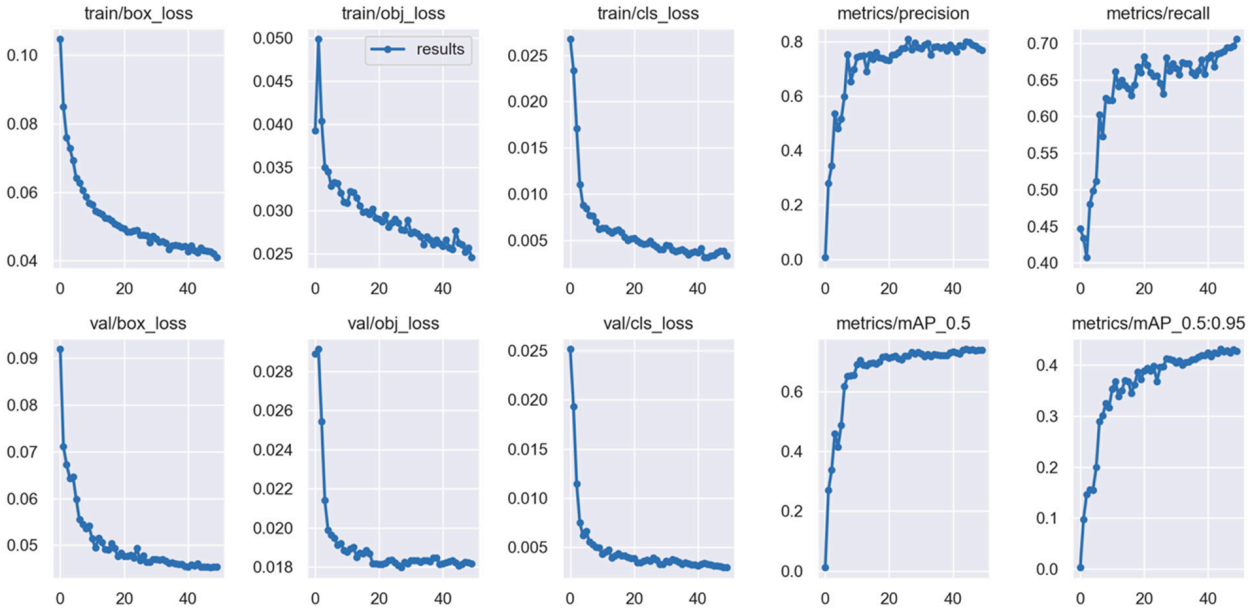


Figure 16. Trash detection model evaluation (Model 2).

The model performed well when tested on the images in Figure 17 which had never been seen by the model before, identifying the dumpsters, trash bags, and cardboard with an accuracy of 84%, 89%, and 84%, respectively. As a result of the model's performance, trash bags and cardboard boxes can be distinguished.

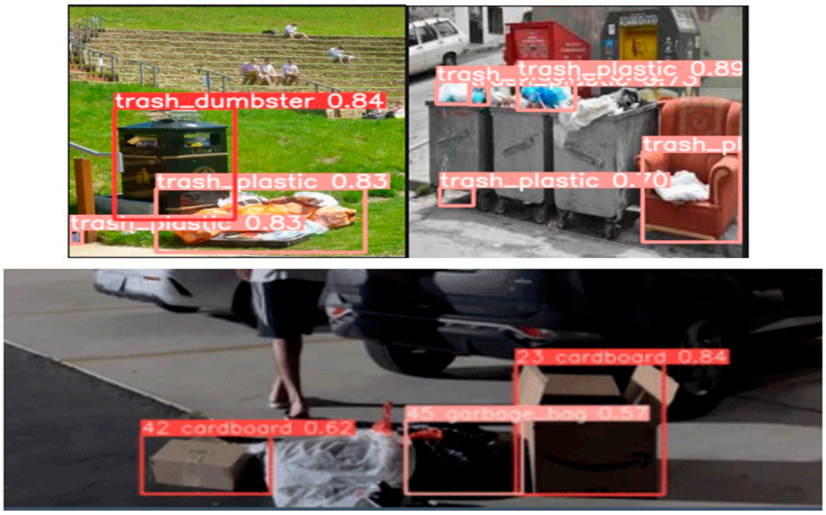


Figure 17. Trash detection model evaluation.

4.5.3. Person Detection Model

A model for person detection was also evaluated based on the mAP and received a score of approximately 93%. The person detection model was implemented as a standalone model in order to improve the accuracy and reduce computation time. The batch size for this model is five and the epochs are 100. The final model's performance evaluation is illustrated in Figure 18.

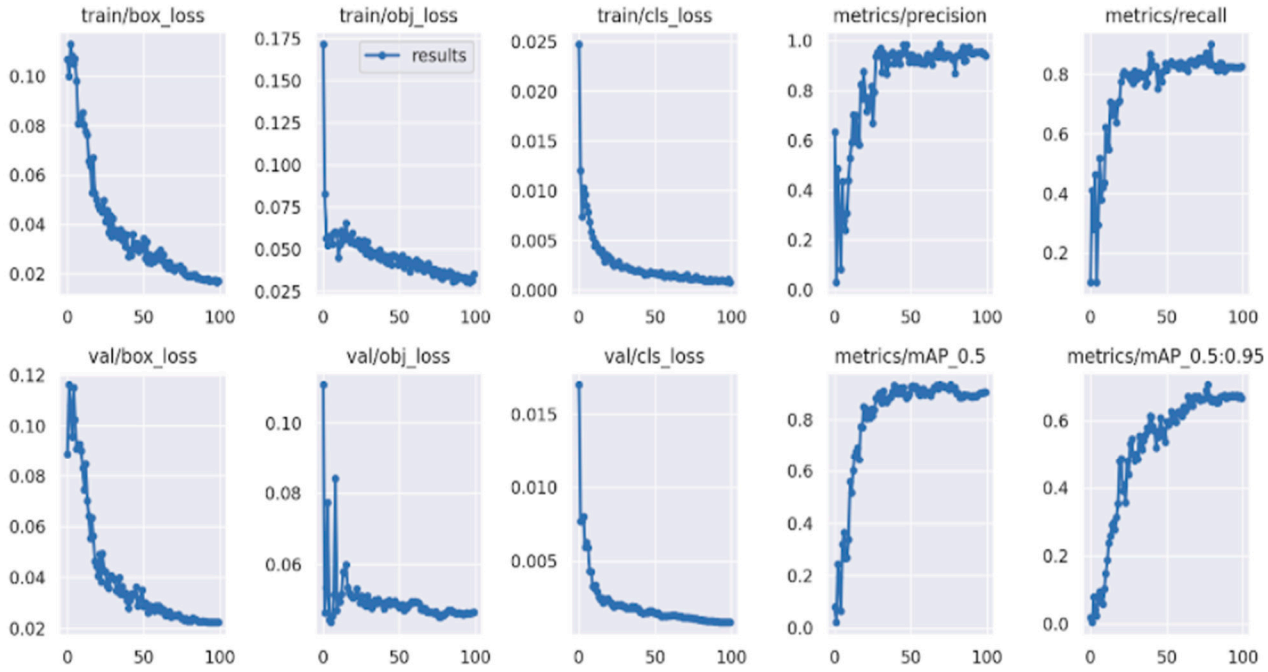


Figure 18. Person detection model evaluation.

The relatively high mAP scores indicate that our model can accurately detect humans, and by using the DeepSORT algorithm, we were able to track each unique person. Figure 17

shows an example of human detection with the model's prediction accuracy (%). Figure 19 shows the initial detection accuracy of 85% for the person detection module and Figure 20 shows the final accuracy of 93% for the integrated module.



Figure 19. Person detection output.

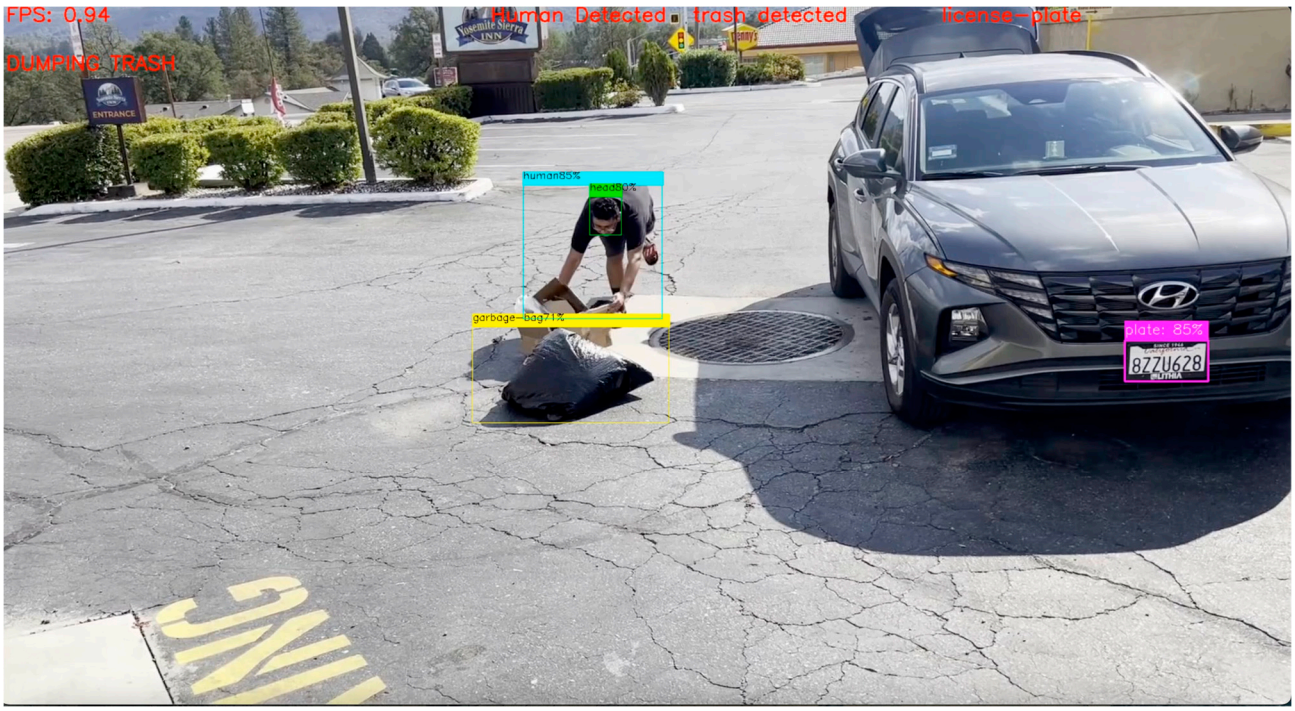


Figure 20. Integrated model results for person, license plate, trash, and illegal dumping action detection.

Figure 20 illustrates the results of integrating all four modules. It provides information about the objects that the model observed, the frames per second (FPSs), and the illegal dumping activity. The detection accuracy of a person and license plate is 85%, while the detection accuracy of a garbage bag is 71%. In addition, the model indicates that the cardboard box is being illegally dumped when the user attempts to dump it.

Tesseract OCR is used to identify the text on the license plate following the detection of the license plate using YOLOv5. In order to read the text properly, the image should be pre-processed by cropping and sharpening the image characteristics at the beginning. The text is then extracted and recognized. As shown in Figure 21, a license plate has been cropped and sharpened, and the text has been extracted for recognition.



Figure 21. Text recognition pre-processing and results.

By calculating the time that it takes for each module to evaluate a single frame or image, we can estimate the runtime performance by counting the number of seconds each module takes to analyze a single frame or image in order to estimate its performance. The likelihood of each classification is used to assess the accuracy of each model, with greater probabilities indicating a more accurate classification. A breakdown of the runtime performance of each module can be found in Table 8. It is possible for runtime performance to vary depending on the processor, which in turn can vary depending on what is available throughout the Colab session. The tracking script calculates the average runtime for the automatic and human detection and tracking modules by first using the YOLO neural network to recognize and classify each object, then using the DeepSORT algorithm to track each classed object individually. First, the car and person detection module assesses the image, and then, the runtime is output for each image. The entire process of identifying and categorizing objects takes approximately 0.039 s, while tracking takes approximately 0.031 s. Detecting a car and a license plate follows the same procedure as detecting a car and a person. As a result of the printout description, it is possible to calculate the average detection time per image, which is 0.032 s after the algorithm has evaluated the data. The progress meter indicates how long it usually takes for the text recognition process to be completed. Show, Attend, and Read (SAR) requires more time to identify the characters in the image than RobustScanner, as shown in Table 8. On average, SAR can detect a single plate image in 0.172 s and RobustScanner does so in 0.14 s. In this case, the time per frame is 0.218 s, of which 0.189 s are used for detection and 0.029 s for tracking.

Table 8. Runtime performance for modules.

Module Name	Hw/Sw Environment	YOLO	DeepSORT	SAR	Total Time
Car and person detection and tracking	Google Colab Pro Processor: Tesla P100-PCIE Memory: 16 GB	0.039 s	0.031 s	NA	0.070 s
Garbage detection and tracking	Google Colab Pro Processor: Tesla P100-PCIE Memory: 16 GB	0.189 s	0.029 s	NA	0.218 s
License plate detection and recognition	Google Colab Pro Processor: Tesla P100-PCIE Memory: 16 GB	0.032 s	NA	0.172 s Robust Scanner: 0.14 s	0.204 s

5. System Development

5.1. System Requirements Analysis

Figure 22 represents the system's implementation of the project. This GUI allows us to input any surveillance video clip, which is then processed by our back-end to produce a text file as the output. The information displayed will include the identity of the perpetrator, a timestamp value indicating when the action occurred, and an OCR-created record of the license plate of the perpetrator, if available. Figure 22 illustrates the design and flow of the GUI. An example of the architecture for web development is shown in this diagram. The first step involves converting the trained model into a .pt file. The model can be hosted locally or in the cloud. The project was developed using Google Colab. To build the front-end user interface of our website, HTML and CSS were used. For the back-end engine, we have selected Flask. Flask has a main function containing the code path that is run only when the HTML submits a POST request. The application will require images and videos to be passed to the UI and the detection will be carried out using PyTorch. The system's implementation flowchart for data storage, documentation of the models and how they are used, computing power to train and operate the models, presentations, and model working demos are all examples of system functional components. Therefore, we chose Amazon S3 as a storage medium for our large dataset. We used Google Colab for preparing, training, and testing the data for our models because it offers a large amount of computational capacity with GPU's runtime. All of our work, updates, and reports are shared through Google Shared Documents. Our system consists of a Nvidia GeForce RTX 2070, an Intel Core i7, and 16GB of system memory, since OpenCV and YOLOv5 require the CUDA toolbox. When an image or video is provided to the system, YOLOv5 and DeepSORT begin to detect and monitor items and persons to identify illegal dumping.

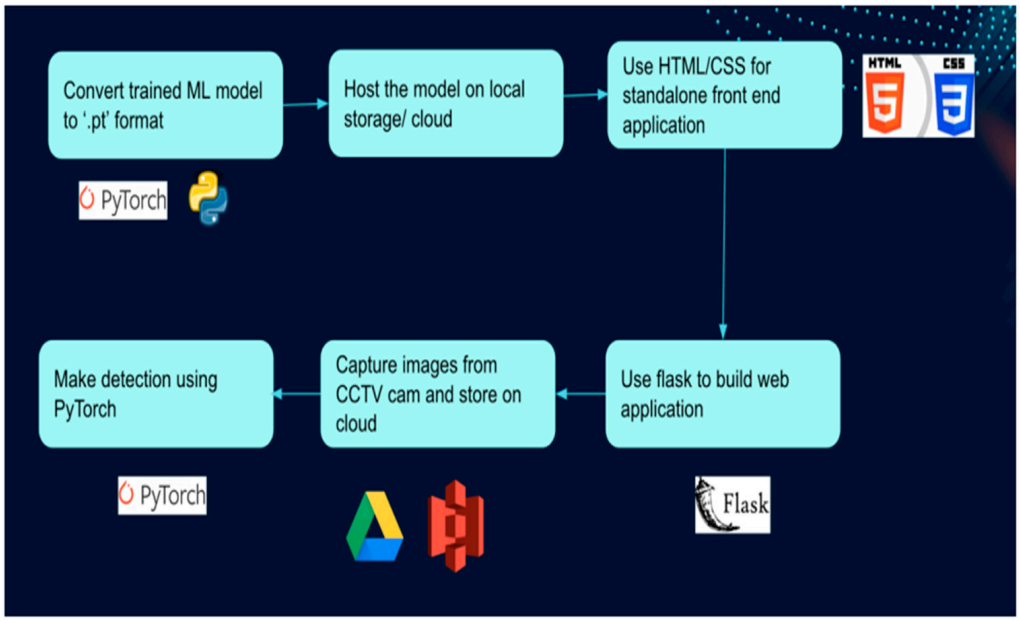


Figure 22. GUI flow of illegal dumping detection.

5.2. System Design

This project involved the implementation of a real-time alert system that sends an email notification to the security team whenever any illegal dumping activity is observed. The email contains a timestamp of the activity, an image of the perpetrator, an image of the vehicle, and the license plate number of the vehicle. The client database stores these details. An interface is created that takes surveillance video as input. To detect illegal dumping in the back-end, a machine learning model is applied to the video input. In the event that

the output is identified, the dumping information is input to the configured database. An image of the perpetrator, a timestamp of the event, a video ID, and an OCR-created record of the license plate of the perpetrator's vehicle are included in the information. Figure 23 illustrates the GUI and Figure 24 illustrates the system's interface.

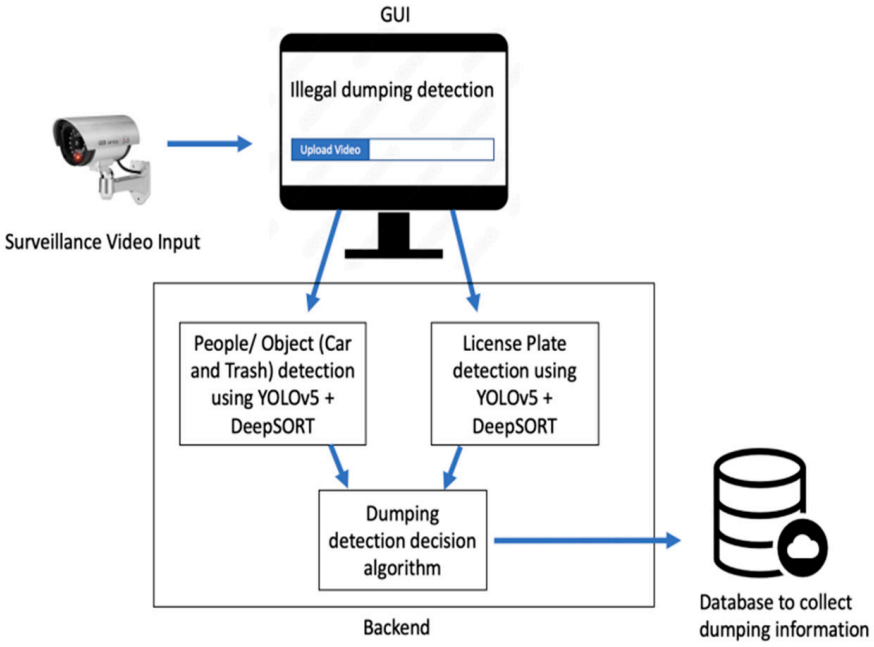


Figure 23. System design.

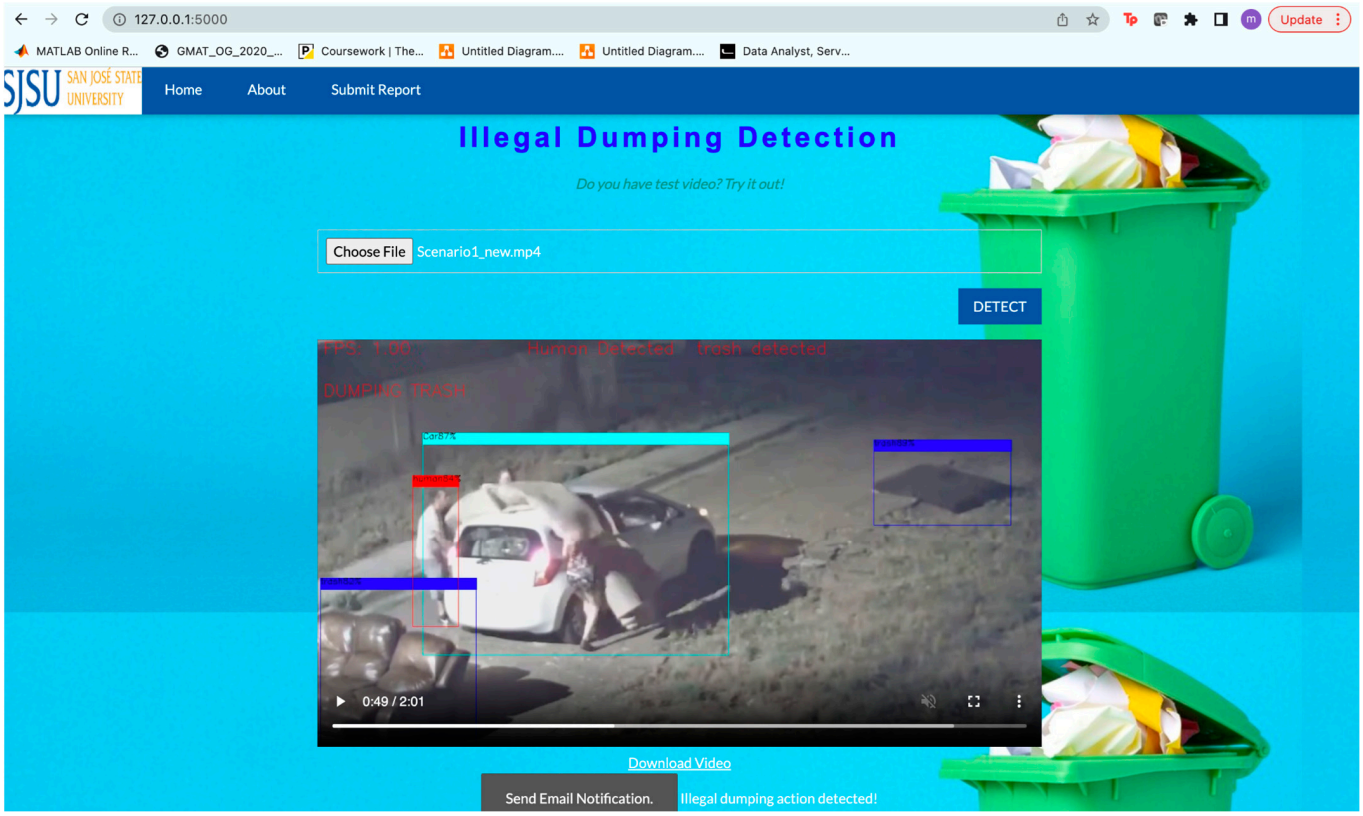


Figure 24. Screenshot of system's interface.

6. Conclusions

6.1. Summary

Illegal dumping has become a serious problem. In some states, such as California, fines of up to USD 10,000 have been imposed. By utilizing machine learning and deep learning technologies, we have been able to solve this problem. We divided the problem into three parts, which are object detection (primarily garbage objects), action detection, and license plate recognition. Each model has been iterated several times (more than 300 times) in order to find the best possible model. By using mAP, we were able to train and evaluate each YOLOv5 model and achieve an overall accuracy of 97%. Each model was equipped with a DeepSORT algorithm for assigning IDs to distinct classes. After that, we developed a simple system of dumping detection based on trash halting within a frame of time. The closest vehicle or person is responsible for dumping the trash, and the information is output in a text file. Flask incorporates the entire framework into its GUI. We found that our study was successful in identifying illegal dumping in surveillance camera frames.

6.2. Recommendations for Future Work

In the future, real-time streaming videos may be monitored in order to assist users in receiving consistent reports and identifying the perpetrators. Infrared cameras with high resolution can be installed to improve the quality of night images. By utilizing an alert system, the project will greatly assist the community in reducing and handling illegal dumping easily without the need for constant monitoring. By doing so, we will be able to reduce the cost and the amount of manpower required. Furthermore, our project reduces dumping on streets and in communities as well as improving the health of society. Our work may inspire others to undertake similar projects in the future and implement the recommendations we have made.

Author Contributions: Conceptualization, N.P., G.B., M.G., V.M., P.S. and J.G.; Methodology, N.P., G.B., M.G., V.M., P.S. and J.G.; Software, N.P., G.B., M.G., V.M. and P.S.; Validation, N.P., G.B., M.G., V.M. and P.S.; Formal analysis, N.P., G.B., M.G., V.M., P.S., F.L. and J.G.; Investigation, N.P., G.B., M.G., V.M. and P.S.; Resources, N.P., G.B., M.G., V.M. and P.S.; Data curation, N.P., G.B., M.G., V.M. and P.S.; Writing—original draft, N.P., G.B., M.G., V.M. and P.S.; Writing—review & editing, N.P., G.B., M.G., V.M. and P.S.; Visualization, N.P., G.B., M.G., V.M. and P.S.; Supervision, F.L. and J.G.; Project administration, F.L. and J.G.; Funding acquisition, F.L. and J.G. All authors have read and agreed to the published version of the manuscript.

Funding: This study was funded in part by Shanxi Province Science and Technology Cooperation and Exchange Special Project Grant No. 202304041101035.

Institutional Review Board Statement: Not applicable.

Data Availability Statement: Publicly available datasets were analyzed in this study. These data can be found here in the references listed [21,22,24,36–39].

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Ebi, K. 6 Cities That Are Fighting Trash with Technology (and Winning!) | Smart Cities Council. 2016. Available online: https://www.smartcitiescouncil.com/article/6-cities-are-fighting-trash-technology-and-winning (accessed on 24 August 2023).
2. Devesa, M.R.; Brust, A.V. Mapping Illegal Waste Dumping Sites with Neural-Network Classification of Satellite Imagery. *arXiv* **2021**, arXiv:2110.08599.
3. Karale, A.; Kayat, W.; Shiva, A.; Hopkins, D.; Nenkova, A. Cleaning Up Philly's Streets: A Cloud-Based Machine Learning Tool to Identify Illegal Trash Dumping. Available online: https://fisher.wharton.upenn.edu/wp-content/uploads/2019/06/PhillyTrash.pdf (accessed on 10 May 2022).
4. Akula, A.; Shah, A.K.; Ghosh, R. Deep Learning Approach for human action recognition in infrared images. *Cogn. Syst. Res.* **2018**, *50*, 146–154. [CrossRef]
5. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016. [CrossRef]

6. Bae, K.; Yun, K.; Kim, H.; Lee, Y.; Park, J. Anti-litter surveillance based on person understanding via multi-task learning. In Proceedings of the British Machine Vision Conference (BMVC), Virtual Event, 7–10 September 2020; pp. 1–13.
7. Yun, K.; Kwon, Y.; Oh, S.; Moon, J.; Park, J. Vision-based Garbage Dumping Action Detection For Real-World Surveillance Platform. *ETRI J.* **2019**, *41*, 494–505. [CrossRef]
8. Matsumoto, S.; Takeuchi, K. The effect of community characteristics on the frequency of illegal dumping. *Environ. Econ. Policy Stud.* **2011**, *13*, 177–193. [CrossRef]
9. Youme, O.; Bayet, T.; Dembele, J.M.; Cambier, C. Deep learning and remote sensing: Detection of dumping waste using UAV. *Procedia Comput. Sci.* **2021**, *185*, 361–369. [CrossRef]
10. Begur, H.; Dhawade, M.; Gaur, N.; Dureja, P.; Gao, J.; Mahmoud, M.; Huang, J.; Chen, S.; Ding, X. An edge-based smart mobile service system for illegal dumping detection and monitoring in San Jose. In Proceedings of the 2017 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computed, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI), San Francisco, CA, USA, 4–8 August 2017. [CrossRef]
11. Coccoli, M.; De Francesco, V.; Fusco, A.; Maresca, P. A cloud-based cognitive computing solution with interoperable applications to counteract illegal dumping in Smart Cities. *Multimed. Tools Appl.* **2021**, *81*, 95–113. [CrossRef]
12. Dahi, I.; El Mezouar, M.C.; Taleb, N.; Elbahri, M. An edge-based method for effective abandoned luggage detection in complex surveillance videos. *Comput. Vis. Image Underst.* **2017**, *158*, 141–151. [CrossRef]
13. Sulaiman, N.; Jalani, S.N.H.M.; Mustafa, M.; Hawari, K. Development of Automatic Vehicle Plate Detection System. In Proceedings of the 2013 IEEE 3rd International Conference on System Engineering and Technology, Shah Alam, Malaysia, 19–20 August 2013. [CrossRef]
14. Torres, R.N.; Fraternali, P. Learning to identify illegal landfills through scene classification in aerial images. *Remote Sens.* **2021**, *13*, 4520. [CrossRef]
15. Sarker, N.; Chaki, S.; Das, A.; Alam Forhad, S. Illegal trash thrower detection based on HOGSVM for a real-time monitoring system. In Proceedings of the 2021 2nd International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST), Dhaka, Bangladesh, 5–7 January 2021. [CrossRef]
16. Carolis, B.D.; Ladogana, F.; Macchiarulo, N. Yolo TrashNet: Garbage detection in video streams. In Proceedings of the 2020 IEEE Conference on Evolving and Adaptive Intelligent Systems (EAIS), Bari, Italy, 27–29 May 2020. [CrossRef]
17. Pedropro. Pedropro/Taco: Trash Annotations in Context Dataset Toolkit. GitHub. n.d. Available online: https://github.com/pedropro/TACO (accessed on 23 March 2022).
18. Srivastava, S.; Divekar, A.V.; Anilkumar, C.; Naik, I.; Kulkarni, V.; Pattabiraman, V. Comparative analysis of Deep Learning Image Detection Algorithms. *J. Big Data* **2021**, *8*, 66. [CrossRef]
19. Laroca, R.; Cardoso, E.; Lucio, D.; Estevam, V.; Menotti, D. On the cross-dataset generalization in license plate recognition. In Proceedings of the 17th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, Virtual Event, 6–8 February 2022. [CrossRef]
20. Laroca, R.; Severo, E.; Zanlorensi, L.A.; Oliveira, L.S.; Gonçalves, G.R.; Schwartz, W.R.; Menotti, D. A robust real-time automatic license plate recognition based on the YOLO detector. In Proceedings of the International Joint Conference on Neural Networks (IJCNN), Rio de Janeiro, Brazil, 8–13 July 2018; pp. 1–10.
21. Common Objects in Context, COCO. Available online: https://cocodataset.org/#home (accessed on 13 May 2022).
22. Trash Annotations in Context, tacodataset.org. Available online: http://tacodataset.org/ (accessed on 13 May 2022).
23. Yolo Is Back! Version 4 Boasts Improved Speed and Accuracy. Synced. 13 August 2020. Available online: https://syncedreview.com/2020/04/27/yolo-is-back-version-4-boasts-improved-speed-and-accuracy/ (accessed on 23 September 2022).
24. UFPR-3DFE Dataset—Laboratório Visão Robótica e Imagem. Laboratório Visão Robótica e Imagem—Laboratório de Pesquisa ligado ao Departamento de Informática, 24 September 2019. Available online: https://web.inf.ufpr.br/vri/databases/uf (accessed on 13 May 2022).
25. Larxel. Car License Plate Detection. Kaggle. 1 June 2020. Available online: https://www.kaggle.com/andrewmvd/car-plate-detection (accessed on 23 March 2022).
26. Sanyam. Understanding Multiple Object Tracking Using DeepSORT. LearnOpenCV. 11 November 2022. Available online: https://learnopencv.com/understanding-multiple-object-tracking-using-deepsort/ (accessed on 23 September 2022).
27. Dabholkar, A.; Muthiyan, B.; Srinivasan, S.; Ravi, S.; Jeon, H.; Gao, J. Smart illegal dumping detection. In Proceedings of the 2017 IEEE Third International Conference on Big Data Computing Service and Applications (BigDataService), Redwood City, CA, USA, 6–9 April 2017. [CrossRef]
28. Optical Character Recognition (OCR): Definition & How to Guide. n.d. Available online: https://www.v7labs.com/blog/ocr-guide (accessed on 25 September 2022).
29. Mccarthy, J. Object Tracking with Yolov5 and Sort. Medium. 27 July 2021. Available online: https://medium.com/@jarrodmccarthy12/object-tracking-with-yolov5-and-sort-589e3767f85c (accessed on 23 September 2022).
30. Liu, Y. An improved faster R-CNN for object detection. In Proceedings of the 2018 11th International Symposium on Computational Intelligence and Design (ISCID), Hangzhou, China, 8–9 December 2018. [CrossRef]
31. Meel, V. Yolov3: Real-Time Object Detection Algorithm (Guide). Viso.ai. 2 January 2023. Available online: https://viso.ai/deep-learning/yolov3-overview/ (accessed on 23 September 2022).

32. Orac, R. What's New in Yolov4? Medium. 24 August 2021. Available online: https://towardsdatascience.com/whats-new-in-yolov4-323364bb3ad3#:~:text=YOLO%20recognizes%20objects%20more%20precisely,boundary%20box%20around%20the%20object (accessed on 23 September 2022).
33. Rath, S. Yolov5—Fine Tuning & Custom Object Detection Training. LearnOpenCV. 28 November 2022. Available online: https://learnopencv.com/custom-object-detection-training-using-yolov5/#:~:text=YOLO%20is%20short%20for%20You,are%20four%20versions%25 (accessed on 25 September 2022).
34. Evaluating Object Detection Models Using Mean Average Precision. KDnuggets. n.d. Available online: https://www.kdnuggets.com/2021/03/evaluating-object-detection-models-using-mean-average-precision.html#:~:text=To%20evaluate%20object%20detection%20models,model (accessed on 25 September 2022).
35. Majumder, S. Object Detection Algorithms-R CNN vs. Fast-R CNN vs. Faster-R CNN. Medium. 4 July 2020. Available online: https://medium.com/analytics-vidhya/object-detection-algorithms-r-cnn-vs-fast-r-cnn-vs-faster-r-cnn-3a7bbaad2c4a (accessed on 23 September 2022).
36. Bileschi, S. CBCL StreetScenes Challenge Framework. 27 March 2007. Available online: https://web.mit.edu/ (accessed on 13 May 2022).
37. The Virat Video Dataset. VIRAT Video Data. n.d. Available online: https://viratdata.org/#getting-data (accessed on 23 March 2022).
38. kevinlin311tw. Kevinlin311tw/ABODA: Abandoned Object Dataset. GitHub. n.d. Available online: https://github.com/kevinlin311tw/ABODA (accessed on 23 March 2022).
39. Kinetics. Deepmind. n.d. Available online: https://deepmind.com/research/open-source/kinetics (accessed on 23 March 2022).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.