Containers in Montevideo: a Multi source Image Dataset

Rodrigo Laguna and Guillermo Moncecchi

Instituto de Computación, Facultad de Ingeniería, Universidad de la República, Montevideo, Uruguay.
{rodrigo.laguna,gmonce}@fing.edu.uy

Abstract. This work presents Clean-Dirty Containers in Montevideo (CDCM), a novel dataset for detection and classification of residue containers. Images were collected from several sources, including Google Street View, Social Networks and smarthpone taken photos. The dataset is publicly available under a Creative Commons License.

Keywords: Multi source Dataset · Smart Cities · Open Data · Image Recognition

1 Introduction

The city of Montevideo, capital of Uruguay, has a population of about 1.5 million people. The main waste collection system for the city is a network of circa 10,000 lateral containers, served by about 45 trucks that perform the collection with different frequencies according to predefined circuits. For several years, one of the main problems in the city has been the appearance of residues around the waste containers, originated from two main causes: sporadic collection problems (due to high demand [10], or occasional union conflicts [8]) and, unfortunately, the growing number of people inadequately sorting waste in the public space [2], or homeless people searching for food for themselves or their animals [11]. This situation occurs heterogeneously in different city zones, depending on the type of residues and the economic and social composition of the population in each area [1].

The Municipality of Montevideo (Intendencia de Montevideo - IM) has taken different approaches to the problem, including a recently announced system to extend the mechanism for immediate complaint attention [12], or the daily adjustment of collection routes based on evidence, seeking to reply faster to the problem. In this context, automatic monitoring of the containers through the city's existing cameras for traffic control or security, or even for cleaning control, currently done manually, could be very useful for a better performance. An automatic waste container recognition and classification system based on image recognition (as opposed to IoT-based systems [3, 4, 9], that require specific hardware for each container) could be used for complaint management, as well as for improve collection mechanisms or container location. To be useful, such a system should recognize the presence of containers in different conditions, since its data sources would be different types of cameras, oriented in different ways, capturing containers from different angles and distances.

One of the most expensive, yet often underestimated, tasks in the automatic image recognition pipeline is the development of an adequate learning dataset. This dataset must include a large enough number of training images, allowing the creation of robust prediction models. In this paper we present a dataset built from different sources designed to be used in two main image recognition tasks: the identification of a lateral container of the city of Montevideo and its classification according to its cleanliness status[1].

The dataset is built on Montevideo's residue collection system. However, several cities in the world, such as, for example, Maldonado (Uruguay), Buenos Aires (Argentina), Barcelona (Spain) or Gijón (Spain), use exactly the same container system. Therefore, models based on the CDCM dataset could be used in any of those cities.

In section 2 we present each of the data sources and their associated image extraction process. Section 3 describes the annotation process to differentiate clean and dirty containers. In section 4 we present some baselines for both tasks. Finally, in section 5 we present some conclusions and further work. All the code involved in this work is publicly available in an open format.[2].

2 Data Sources

This section reports the dataset sources and describes the process we followed to acquire them. Since the goal was to build a multi-purpose dataset, we considered images from a wide and distinct range of sources. We provide a metadata file, summarized in Table 1, containing each image source, and its coordinates as *(latitude, longitude)* pairs when available.

2.1 Container photos

Out first approach was very simple: we took photos of different containers within the city. To increase its number and variability, we asked people on social networks to contribute images of containers from their neighbourhoods. Images in this subset include photos taken from different perspectives and camera types, including also images taken from a moving vehicle. Note that the same container could appear more than once, visualized from different perspectives, or in different days.

[1] Dataset collection and annotation was carried out between May 2020 and February 2021, with six incremental versions during this period of time, all of them available at `https://www.kaggle.com/rodrigolaguna/clean-dirty-containers-in-montevideo`. This paper is based on version 6.1 of the dataset.

[2] Code: `https://github.com/rola93/clean-dirty-preprocess-baseline`

Table 1. Image sources. The row *other* refers to images coming from social networks, Google images and local media.

	Train		Test		
Sources	Clean	Dirty	Clean	Dirty	Total
GSV	352	109	30	2	493
PorMiBarrio	10	29	28	311	378
Container photos	815	716	540	176	2247
Other	29	97	2	106	234
Total	1206	951	600	595	3352

2.2 Google Street View images

To increase the dataset, we decided to include images from Google Street View (GSV). For all cities in Uruguay, those images were taken in 2015. Since the IM, as a result of its open data policy [5, 15], regularly publishes the location of every single container in the city [6], we used it as a starting point to query GSV API on certain locations, trying to grab an image from the corresponding container.

Fig. 1. This figure shows all eight extracted images corresponding to a single location point. Red text under each image shows its labels in the Garbage Containers in Montevideo dataset, built out of GSV images.

To extract reasonably good images, the size was manually fixed at 600x600 pixels, the location was provided as latitude longitude pairs, the field of view (FOV) was fixed at 120, and the pitch was 0. The *heading* parameter is very important[3], since it defines the compass heading of the camera, mapped to cardinal points: values range from 0 to 360, both indicating North; 90 for East, and 180 South. Since we aimed to include different views and we did not know exactly where the containers were, each point was retrieved with 8 different heading values, increasing 45 degrees on each.

After this initial phase, we had eight images for each of the 4200 containers positions provided by IM. However, most of those images do not actually include a container, due to two factors: first, containers are periodically moved and, given the mismatch between the GSV acquisition date and the IM dataset publication, reported locations could differ from the actual location. To mitigate this effect, the location map version closest to the date of the GSV data collection, dated December 2017, was used. Second, we do not know in which direction the container is pointing. We get 8 different images for each point, but a container can be present in up to three of them. Figure 1 show some examples.

Since manually checking which of the 33600 downloaded images actually included a container would be too time-consuming, we trained a classifier on a small dataset of about 3500 of those images, and then used it to classify the rest of them (see Table 2 for details). We named this small dataset as Garbage Containers in Montevideo.[4] It addresses the problem as a binary classification task: whether or not the image actually includes a container. The classifier was built using transfer learning from a MobileNetV2 architecture [14] trained over ImageNet. We replaced its last layer with three layer blocks, each containing a fully connected layer followed by dropout, and a single output neuron with sigmoid activation. This classifier achieved an accuracy of 0.86 on the test dataset and 0.89 F-1 score for the container class.

With this classifier we selected those images classified as containers, until we got 2500 images. Those 2500 images were manually labelled as clean or dirty, and included in the dataset. Note that most of the images were discarded since containers were not completely visible or they were too far in the image. We did not include all GSV images to avoid those images from being over-represented in the dataset. We took this approach since GSV images have some “non realistic” features: they contain several watermarks from Google, while some other parts are blurred. Table 3 shows how were them distributed in the CDCM dataset.

Table 2. GSV-extracted Containers in Montevideo dataset

Class	Training	Test	**Total**
Container	938	284	**1222**
No Container	1750	516	**2266**
Total	**2688**	**800**	**3488**

[3] GSV API parameters documentation: `https://developers.google.com/maps/documentation/streetview/overview#url-parameters`

[4] This dataset is also publicly available under a Creative Commons Licence at `https://www.kaggle.com/rodrigolaguna/garbage-containers-images-in-montevideo`

Table 3. GSV images in each split of the CDCM dataset. Note that most instances are of clean containers, reflecting the fact that dirty containers in Montevideo tend to be more frequent in particular days, while regularly they are clean.

Class	Training	Test	**Total**
Clean	352	30	**382**
Dirty	109	2	**111**
Total	**461**	**32**	**493**

2.3 Social Networks

Seeking to increase diversity on the dataset, the next step we took was to consider social networks. In this context, social networks proved very useful as an information source. When containers have cleanliness problems, people complain about, especially in Twitter and Facebook. What is more, sometimes those overflown containers events are covered by local media. In reply to those complaints, when authorities take action on them, they often reply with images showing the same containers after being cleaned.

To find and add those type of images to the dataset, we implemented two closely related strategies: we queried Twitter, Facebook and Google for certain words such as *"mugre"*, *"sucio"*, *"contenedor"* or *"basura"* [5] and we downloaded their associated images. Additionally, we checked local authorities' social network profiles, and scroll over its multimedia content looking for additional (and probably clean) containers. Those images are usually extreme cases: claims for too dirty containers or reply with completely clean ones by the authorities.

2.4 PorMiBarrio

PorMiBarrio[6] is a web application where citizens can complain for incidents and problems in the city. Users of the application report those incidents with a brief description and they can also include some photos showing the problem. This information is sent to local authorities.

Data Uruguay,[7] the civil organization behind PorMiBarrio, provided us with 2400 images from different reports, belonging to many different categories. Those images were manually labelled to keep only those depicting a container, and labelled as clean or dirty. This process ended up with about 400 additional images, most of them corresponding to dirty containers.

3 The CDCM Dataset

After collecting the images, we labelled the dataset for two independent yet related tasks. First, each image is labelled with a bounding box indicating the

[5] Spanish words for *"grim"*, *"dirty"*, *"container"*, *"garbage"*
[6] `https://pormibarrio.uy/`.
[7] `https://data.org.uy/`.

position of the containers within the image. Images contain from one to five containers. Second, we manually labelled each image with a single label: clean or dirty, according to the cleanliness of the containers in the image, characterising container cleanliness as a binary image classification problem. Detaching the container localization and the clean/dirty status allows to potentially include new tasks such as, for example, open/closed containers, the need of maintenance or to detect vandalized containers.

3.1 Training test splits

Despite container detection and clean/dirty classification being independent, both tasks share exactly the same images, and what is more, they use the same training test split.

The dataset contains 3412 images, with 35 % being used for testing. Containers are often captured multiple times with small variations. For instance, in Figure 1 the same container is present in three images with different perspectives. This also happen when dirty containers are reported to authorities in social network and for manually taken images, not only with differences in perspectives but also with a single container appearing in different days. So we provide independent splits to make sure that each container consistently goes either to training or testing split.

Images in the the test split are also more diverse, since our main goal is to have a strong test set. This is why, since images from PorMiBarrio were reported in non-controlled environments and taken in different conditions, they were all included in the test split. Most of the images taken from social networks were also included in the Test split. On the other hand, GSV images were used more frequently in the training test, because they are non realistic: they have several details, like overlay text, blur and distortions which are only common in GSV images. Taking that into account, they were included only to increase the dataset size, since GSV itself is not an interesting scenario to implement a container recognition and classification system. Table 3 shows this distribution.

3.2 Container Detection

The containers appear, within each image, in different positions: sometimes the container occupies the whole image, while in other cases, it appears in just a small region of it (Figure 2 shows eight different cases). In addition to this, residues are only relevant when it is surrounding a container. Even the container's position within the image is a priori unknown, and must be found before classifying the container into clean or dirty, or performing any other task. To tackle those problems, the position of each container in the image is provided as bounding boxes, following Pascal VOC annotation format. This allows to face the task of Container Position Identification as an Object Detection problem. The bounding boxes just include the container limits, not including any residues spread around.

Fig. 2. Eight different images showing a wide range of perspectives for different containers.

Table 4. Number of containers per image in the dataset. Each column shows the number (and portion) of images containing N containers on it, with N in $[1, \ldots, 5]$. Note that there is just one image with five containers on it in the training set, and only two in the test set. In addition to this, the test split contains a larger portion of images with a single container than the train split.

		1 container	2 containers	3 containers	4 containers	5 containers	Total
Train	images	1725	454	34	3	1	2217
	portion	77.28 %	20.48 %	1.53 %	0.14 %	0.05 %	100.0 %
Test	images	1015	154	20	4	2	1195
	portion	84.94 %	12.89 %	1.67 %	0.33 %	0.17 %	100.0 %

The dataset contains 2217 images for training and 1195 for testing, each containing between one and five containers (Table 4). Both splits contain far more images with a single container on it.

Containers appear in different portions of the image, going from 3.0 % of its pixels to 91.7 % of its pixels for training images. For testing images, the range of pixels portion is similar and goes from 8 % to 99.9 % of the image. For reference, in training images, containers occupy 15.2 ± 12.9 % of pixels, while for test images 18.4 ± 13.4 % (mean ± standard deviation). The difference can be explained by how those splits are composed: as exposed in Section 2, GSV images, which has several images where containers are not the main point, are mainly used for training, while images from PorMiBarrio are used for testing, and most of the time containers occupy a relevant part of the image.

3.3 Clean/Dirty Classification

Fig. 3. This figure shows containers with different residues around them. The labels are included in red or green texts at the bottom of each image, for dirty and clean classes respectively.

The key feature of the dataset consists in differentiating clean from dirty containers. One problem is that there is not an objective definition on what a dirty container is. It is different to have just a few boxes out of the container than an important number of residues wide spread around an overflown container.

Figure 3 illustrates some of those cases. Those extremely different situations also require different actions from authorities to solve them. Binarizing such complex reality implies that a wide range of situations must be mapped to either clean or dirty. In this section we describe the criteria used to differentiate both cases.

Clean and dirty containers were labelled at image level, according to the global situation in each image. This allows to model the problem as an image classification task. This implies that images containing several containers in it, could have a clean container and a dirty one, each with its bounding box as previously described in Section 3.2, but the complete image is labelled as dirty.

Labeling the clean dirty status at image level remove the problem of defining up to which point residues on the image belong to the container or not. What is more, when more than one container is present on image, to define which is the dirty one and which the clean is not important.

Table 5. Number of images for each class in the CDCM dataset.

Classes	Train	Test	**Total**
Clean	1206	600	**1806**
Dirty	1011	595	**1606**
Total	**2217**	**1195**	**3412**

It is hard to know beforehand the actual distribution of clean and dirty containers in the city. The overflown events due to lack of service are expected to be infrequent in the city, and mainly respond to events with higher residue generation, such as Christmas, or to union conflicts. However, scattered residue events are more common, and they had increased during the last year with the economic crisis derived associated to the COVID-19 pandemic: there are more people digging in the containers looking for valuable residues or even food. Since it is difficult to know *a priori* this distribution, we decided to balance the number of clean and dirty containers in the dataset. The actual distribution of instances in the dataset can be seen in Table 5.

4 Baseline models

To first evaluate the built dataset on the two separate tasks of container detection and cleanliness classification, we built simple baselines, following again a transfer learning approach.

4.1 Container Detection

To build the baseline model for detection task we used ImageAI [7], a Keras' based open source library for computer vision tasks. ImageAI provides ready to use interfaces to train object detection models. We used YoloV3[13] as the base architecture, with weights trained in COCO dataset. We trained this model for

30 epochs, with default parameters. The model built achieves a mean average precision (mAP) of 0.56 in the test set.

4.2 Classification

For cleanliness classification, the baseline model is a MobileNetV2 [14] architecture, pretrained on ImageNet, implemented in Keras [4]. Weights are used just as feature extractors and, therefore, are fixed during all the training process. We stacked a fully connected layer, with 1024 neurons on it and ReLU as the activation function, and including also a Dropout mechanism with a rate of 0.5. The output layer contains a sigmoid activated single neuron. We used data augmentation for training and testing, and we used early stopping to avoid overfitting. Since our main goal was to establish a baseline, we used the test dataset to implement the early stopping, which may lead to overestimated results. However, this is mitigated by the fact that data augmentation was also applied to test set. For data augmentation, both train and test used random on line transformations: horizontally flips, random rotating up to 15 degrees, up to 10 % of zoom, 10 % of width and height shifts and up to 10 % of shear range.

Table 6 summarizes the performance measures. Note that the full image is being classified even when, in several cases, only a small region of the image is relevant for the task.

Table 6. Baseline model results. Loss values corresponds to cross entropy between true labels and predicted labels.

Split	Accuracy	Loss
Training	83.81	0.360
Test	82.51	0.366

5 Conclusions and future work

This work presented a novel dataset, built from different information sources, intended to help to monitor the state of residues containers, through automatic image recognition based on regular camera images. The dataset is labelled for two close related tasks, container detection and cleanliness classification, for the city of Montevideo, Uruguay. The whole dataset is fully available under a Creative Commons licence.

Since we have collected an important number of images from different sources, we are considering further labelling the dataset, including more classes or even a cleanliness/dirtiness score. In addition to this, and to improve performance, we plan to further improve and document the annotation criteria, and use multiple annotators to label each image, improving the dataset quality.

References

[1] *¿Cómo se limpia Montevideo?* Oct. 2020. URL: https://montevideo.gub.uy/areas-tematicas/ambiente/gestion-de-residuos/como-se-limpia-montevideo (visited on 06/22/2021).

[2] La Diaria. "Hay 16% más personas en situación de calle en Montevideo respecto de 2020, según el Mides". In: *La Diaria* (Aug. 4, 2021). URL: https://ladiaria.com.uy/politica/articulo/2021/8/hay-16-mas-personas-en-situacion-de-calle-en-montevideo-respecto-de-2020-segun-el-mides/ (visited on 08/20/2021).

[3] Umar Draz et al. "A real-time smart dumpsters monitoring and garbage collection system". In: *2017 Fifth International Conference on Aerospace Science & Engineering (ICASE)*. IEEE. 2017, pp. 1–8.

[4] Chizea D Francis and Akachukwu M Chichebe. "Smart Waste-Bin Monitoring System". In: *Engineering and Science* 4.3 (2019), pp. 61–63.

[5] Guillermo Moncecchi. *Towards a public digital infrastructure: why do governments have a responsibility to go open?* Dec. 2012. URL: https://blog.okfn.org/2012/11/01/towards-a-public-digital-infrastructure-why-do-governments-have-a-responsibility-to-go-open/ (visited on 06/22/2021).

[6] Intendencia de Montevideo. *Estado de recolección de contenedores de residuos domiciliarios*. Dec. 2017. URL: https://catalogodatos.gub.uy/dataset/intendencia-montevideo-estado-de-recoleccion-de-contenedores-de-residuos-domiciliarios (visited on 06/22/2021).

[7] Moses and John Olafenwa. *ImageAI, an open source python library built to empower developers to build applications and systems with self-contained Computer Vision capabilities*. Mar. 2018. URL: https://github.com/OlafenwaMoses/ImageAI.

[8] El Observador. "Pulseada entre Adeom y la IMM por la esencialidad en la recolección de residuos". In: *El Observador* (Jan. 23, 2019). URL: https://www.elobservador.com.uy/nota/pulseada-entre-adeom-y-la-imm-por-la-esencialidad-en-la-recoleccion-de-residuos-2019122164332 (visited on 08/20/2021).

[9] MF Omar et al. "Implementation of spatial smart waste management system in malaysia". In: *IOP Conference Series: Earth and Environmental Science*. Vol. 37. 1. IOP Publishing. 2016, p. 012059.

[10] El Pais. "IMM dice que ayer hubo atraso en recolección en 25% de los contenedores, pero que hoy mejoró". In: *El Pais* (Dec. 25, 2020). URL: https://www.elpais.com.uy/informacion/servicios/imm-dice-ayer-hubo-atraso-recoleccion-contenedores-hoy-mejoro.html (visited on 08/20/2021).

[11] Montevideo Portal. "Falta es desparramar basura, no dejar una bolsa fuera del contenedor". In: *Montevideo Portal* (Mar. 6, 2020). URL: https://www.montevideo.com.uy/Noticias/La-IM-esta-multando-a-quienes-dejan-basura-fuera-de-los-contenedores-uc746151 (visited on 08/20/2021).

[12] *Primera etapa de Montevideo más verde en línea*. June 2021. URL: https://montevideo.gub.uy/noticias/medio-ambiente-y-sostenibilidad/primera-etapa-de-montevideo-mas-verde-en-linea (visited on 06/22/2021).

[13] Joseph Redmon and Ali Farhadi. "Yolov3: An incremental improvement". In: *arXiv preprint arXiv:1804.02767* (2018).

[14] Mark Sandler et al. "Mobilenetv2: Inverted residuals and linear bottlenecks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 4510–4520.

[15] Fabrizio Scrollini. "Open cities: The case of Montevideo". In: (2014). URL: http://www.opendataresearch.org/content/2014/662/open-cities-case-montevideo.