technologies

MDPI

Article

A Deep Learning-Based Dirt Detection Computer Vision System for Floor-Cleaning Robots with Improved Data Collection

Daniel Canedo *, Pedro Fonseca, Petia Georgieva and António J. R. Neves

Institute of Electronics and Informatics Engineering of Aveiro/Department of Electronics, Telecommunications and Informatics (IEETA/DETI), University of Aveiro, 3810-193 Aveiro, Portugal; pf@ua.pt (P.F.); petia@ua.pt (P.G.); an@ua.pt (A.J.R.N.)

* Correspondence: danielduartecanedo@ua.pt

Abstract: Floor-cleaning robots are becoming increasingly more sophisticated over time and with the addition of digital cameras supported by a robust vision system they become more autonomous, both in terms of their navigation skills but also in their capabilities of analyzing the surrounding environment. This document proposes a vision system based on the YOLOv5 framework for detecting dirty spots on the floor. The purpose of such a vision system is to save energy and resources, since the cleaning system of the robot will be activated only when a dirty spot is detected and the quantity of resources will vary according to the dirty area. In this context, false positives are highly undesirable. On the other hand, false negatives will lead to a poor cleaning performance of the robot. For this reason, a synthetic data generator found in the literature was improved and adapted for this work to tackle the lack of real data in this area. This synthetic data generator allows for large datasets with numerous samples of floors and dirty spots. A novel approach in selecting floor images for the training dataset is proposed. In this approach, the floor is segmented from other objects in the image such that dirty spots are only generated on the floor and do not overlap those objects. This helps the models to distinguish between dirty spots and objects in the image, which reduces the number of false positives. Furthermore, a relevant dataset of the Automation and Control Institute (ACIN) was found to be partially labelled. Consequently, this dataset was annotated from scratch, tripling the number of labelled images and correcting some poor annotations from the original labels. Finally, this document shows the process of generating synthetic data which is used for training YOLOv5 models. These models were tested on a real dataset (ACIN) and the best model attained a mean average precision (mAP) of 0.874 for detecting solid dirt. These results further prove that our proposal is able to use synthetic data for the training step and effectively detect dirt on real data. According to our knowledge, there are no previous works reporting the use of YOLOv5 models in this application.

Keywords: computer vision; deep learning; object detection; floor-cleaning robots

Citation: Canedo, D.; Fonseca, P.; Georgieva, P.; Neves, A.J.R. A Deep Learning-Based Dirt Detection Computer Vision System for Floor-Cleaning Robots with Improved Data Collection. *Technologies* **2021**, *9*, 94. https://doi.org/10.3390/technologies9040094

Academic Editors: Bungo Ochiai, Go Matsuba, Tomoya Higashihara, Sathish K. Sukumaran and Kohei Osawa

Received: 1 November 2021
Accepted: 24 November 2021
Published: 1 December 2021

1. Introduction

Floor-cleaning robots that use digital cameras as a means to detect dirty spots are a relatively new concept. Recently camera based mapping has been explored in floor-cleaning robots, complementing other navigation sensors [1]. Since these floor-cleaning robots already incorporate cameras, it is reasonable to use them for tasks other than navigation. These other tasks mainly revolve around detecting dirty spots. While detecting dirty spots could be useful to tell the robot which direction to go, the literature seems to indicate that researchers are trying to detect dirty spots to save robots' resources [2], to distinguish between solid and liquid dirt [3] or to distinguish between dirt items and useful objects [4].

Grünauer et al. [2] proposed a dirty spot detection system using an unsupervised approach to train Gaussian Mixture Models (GMMs) [5] to represent the floor pattern. The captured image is converted to CIELAB color space [6] resulting in three images, one image for each channel (L*a*b*), which allows for the color information to be separated from the illumination. Afterwards, the gradient of each channel is calculated and the three images are divided into blocks. For each block, the mean and standard deviation are calculated, which will be used to train three GMMs (one per image). In this way, GMMs learn to represent the floor pattern present in a given image. Anything outside of this pattern is considered dirt. However, this approach has several problems: It will not work for partially blurred and/or uneven illuminated images because GMMs will not be able to effectively represent the floor pattern. Another relevant problem is that not everything in a given image that is not within the floor pattern is dirt, but this approach will detect it as such. Nonetheless, the authors show an approach that does not need a separate step of learning whenever the robot faces a new floor pattern, showing a viable solution for floor-cleaning robots.

Ramalingam et al. [3] proposed the use of an architecture that includes components such as Single-Shot MultiBox Detector (SSD) [7], MobileNet [8], and Convolutional Neural Network (CNN) [9] for classifying solid and liquid dirt on the floor. The MobileNet is tasked with extracting features while the SSD is tasked with the detection and classification. This is then paired with a Support Vector Machine (SVM) [10] to classify liquid dirt based on size such that the cleaning device is able to identify regions that are harder to clean. The authors built a custom dataset of 2000 images captured with diverse floor patterns with various classes of solid food waste, semi-solids, and large liquid spillage. To avoid overfitting, they applied Data Augmentation [11] by performing geometrical transformations to the dataset. Then, they used 10-fold cross-validation for the training step. In the test phase, the system obtained an accuracy higher than 96% in classifying both solid and liquid dirt.

Bormann et al. [4] proposed a tool to artificially generate data [12] to tackle the scarcity of data in this area. In addition to this tool, the authors proposed a dirt detection system based on the YOLOv3 framework (You Only Look Once) [13]. YOLO is an object detection algorithm supported by a CNN. This framework was born in response to algorithms such as the Region Based Convolutional Neural Network (RCNN) [14] that needed to propose regions of interest in the image to be analyzed and then pass those regions through a CNN to detect objects. As this process is computationally expensive, the YOLO framework proposed a solution which divides the input image into a grid and, for each block of the grid, a bounding box and its probability of belonging to a certain class are generated. This process is significantly faster than solutions like RCNN because the image only needs to go through the CNN once, not requiring additional steps such as region proposals. In addition to being faster, the YOLO framework has state-of-the-art results measured through the mAP in several benchmarks, such as COCO [15]. Consequently, the paper that proposed the use of YOLOv3 to detect dirt, looking at the problem as an object detection problem, managed to obtain state-of-the-art results on the application of floor-cleaning robots. In this paper, the authors compared their results with the results of the GMMs [2] mentioned in this document, showing better performance.

The main challenges found in the literature in the context of floor-cleaning robots were the following:

- Great variations in light intensity.
- Complex floor patterns.
- Lack of enough labelled images covering various dirty scenarios.
- Blurred images due to robot movement.
- Dirt/Clean discrimination.

In order to tackle the challenges mentioned above, our work takes the tool proposed by [4] and adapts it to generate a synthetic dataset that has a great floor variety and dirt samples. This dataset is used to train on the YOLOv5 framework [16]. Then a real dataset (ACIN) [17] is used to test the trained models. This work tries to tackle the same problem

described in [3] with an adapted and improved solution from [4]. While in [3] the authors state a 96% accuracy in identifying and classifying both solid and liquid dirt, the training and testing datasets are quite similar. Furthermore, their approach is more costly since the cleaning robot needs to capture the training dataset which is manually annotated afterwards. Additionally, their approach requires an extra SVM classifier to classify dirt based on size. YOLOv5 overcomes this problem since it allows for multi-scale predictions.

In this paper a synthetic dataset is generated for the training step. This approach is more practical since one can generate as much labelled data as needed. It is also important to note that the testing dataset consists of real data and is significantly different from the training dataset. These differences come from cameras, camera distances, camera angles, solid dirt, liquid dirt, floors, illumination, backgrounds, and so on.

The main novelty of this paper is achieved during the synthetic dataset generation. Some floor images which have other objects are selected. The floor is then segmented such that dirty spots are only generated on the floor and do not overlap those objects. This helps the models to distinguish between dirty spots and objects, reducing the number of false positives. This synthetic dataset is used for training YOLOv5 models which are then tested on a significantly different and challenging dataset (ACIN). The results shown in this paper further prove that our proposal is able to generalize well on real data even though synthetic data was used for training. The main contributions of this paper are the generation of a synthetic dataset with 2 different types of dirtiness (dirt and liquid); the annotation of the ACIN dataset improving its quality and robustness; the study of the models provided by the YOLOv5 framework when applied to the problem of detecting dirty spots on the floor; the recognition of relevant problems that can be useful for future research. We believe that this work can help the scientific community in this application, therefore the generated dataset, the new annotations of the ACIN dataset, and the updates done to the data generator tool are made public at the end of the document.

This paper is organized as follow: Section 2 presents the methodology; Section 3 presents the results; Section 4 presents the discussion; Section 5 presents the conclusion.

2. Materials and Methods

As mentioned in Section 1, there is a scarcity in annotated data in the area of automated cleaning systems. This scarcity is a problem in Machine Learning approaches because of overfitting [18]. A model overfits when it classifies data used for training accurately, but its accuracy drops significantly when facing new data (poor generalization). In order to tackle this problem, a tool proposed by [4] which is able to generate synthetic data was considered. This tool can blend dirt samples into clean floors in random locations, adding simulated light sources and shadows. It can also increase the number of data by applying geometric transformations to both floors and dirt samples. Therefore, it is possible to generate a large dataset from a small number of images. However, this tool presented a few problems that had to be tackled. Consequently it was upgraded and adapted as follows:

- Outdated code was updated.
- Major bugs were corrected.
- It can now adapt to different image resolutions.
- The dirt size is now automatically scaled for different floor resolutions.
- It can now generate liquid dirt samples by manipulating their transparency.
- It is now prepared to check if the floor images have segmentation masks associated so that the dirt is only generated on the floor.
- The output blended images and labels are now generated in YOLO format and can be directly used by the YOLO framework.

A link to access these changes is provided at the end of this document. Generating the dataset required previously obtaining images of solid dirt, liquid dirt, and clean floors. The Trashnet dataset [19] and Google Search were used to obtain solid dirt samples. These samples were then segmented using the segmentation tool proposed by Marcu et al. [20].

This resulted in 141 solid dirt samples, and Figure 1 illustrates some examples with the respective masks resulting from the segmentation.

Figure 1. Solid dirt samples and the masks resulting from the segmentation.

As for the liquid dirt samples, no dataset was found and therefore only Google Search was used to obtain images. They were equally segmented as the solid dirt samples resulting in 15 liquid dirt samples, and Figure 2 illustrates some examples.

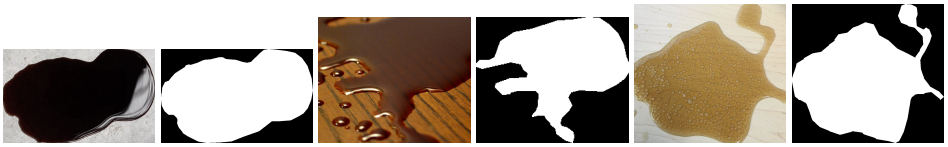

Figure 2. Liquid dirt samples and the masks resulting from the segmentation.

As for clean floors, no dataset was found so Google Search was used to obtain images. The literature seems to indicate that the floor-cleaning robots' cameras are usually placed at 0.6 m to 1.1 m from the ground and have an inclination of 40 degrees to 90 degrees downwards. As for our floor-cleaning robot prototype, the cameras are placed at 0.7 m from the ground and have a configurable inclination of 45 degrees to 70 degrees downwards. Floor images that resemble these conditions were considered. Figure 3 shows our floor-cleaning robot prototype.

Figure 3. Floor-cleaning robot prototype.

Another criteria for choosing floor images was the following: Some images only represented a clean floor, while others had some objects such as walls, doors, chairs, shoes, wires, and so on. These floors that contain objects were added to help the YOLOv5 models distinguishing between them and actual dirty spots. In order to blend solid dirt and liquid dirt only into the floor, floor images that contain objects were segmented. This resulted in 496 types of floor. Figure 4 illustrates clean floors with no objects.

Figure 4. Clean floors with no objects.

Figure 5 illustrates clean floors that contain objects and the masks resulting from the segmentation.

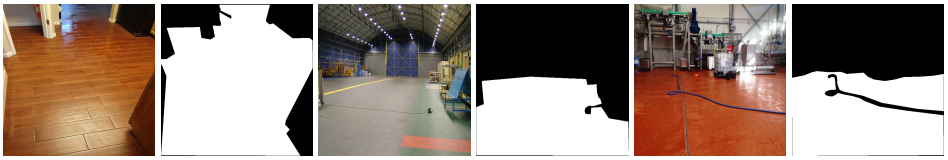

Figure 5. Clean floors that contain objects and the masks resulting from segmentation.

Using the tool for data generation mentioned above, the following strategy was used:

- Number of solid dirty spots per frame: 5, 6 or 7
- Number of liquid dirty spots per frame: 2
- Flip clean floor images horizontally and vertically, generating 4 images out of 1 floor image
- A probability of 1% was set to generate background images, with no blended solid dirt nor liquid dirt.

Figure 6 summarizes the process of generating synthetic data.

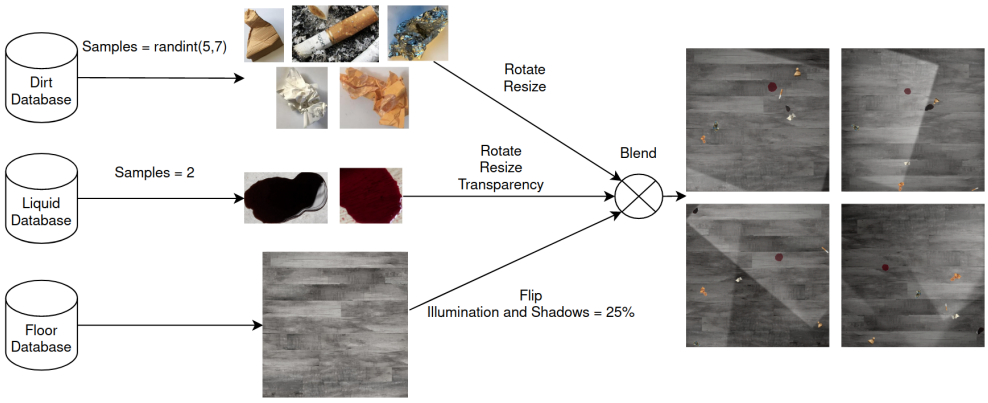

Figure 6. Pipeline of generating synthetic data.

With this, a dataset of 1984 images with 11,727 instances of solid dirt and 3920 instances of liquid dirt was generated and a link to access it is provided at the end of this document. As mentioned in [4], the idea of training models with larger datasets being correlated with better results is not true in this situation since the visual variance of the provided solid dirt and liquid dirt can be covered with smaller datasets.

For a rigorous testing, the ACIN dataset was considered since it consists of real images. This dataset covers most of the challenges mentioned in Section 1, namely severe lighting conditions, complex floor patterns and blurred images. However, out of the 971 images only 248 images were annotated. For this reason, the dataset was annotated from scratch using the tool LabelImg [21]. This resulted in 694 annotated images with 1765 instances of solid dirt and 521 instances of liquid dirt, the remaining images that were not annotated represent clean floors. A link to access this annotation is provided at the end of this document. Figure 7 aims to provide a visual comparison between the generated dataset and the ACIN dataset.

Figure 7. Generated dataset in the first row and ACIN dataset in the second row.

As mentioned in Section 1, the YOLOv5 framework was used. This version of the YOLO family was chosen due to some important improvements over the older versions. For the backbone, YOLOv5 incorporates Cross Stage Partial Network (CSPNet) [22] with Darknet to extract features. This backbone helps dealing with redundant gradient information, reduces the model size, and improves both accuracy and speed. For the neck, YOLOv5 uses the Path Aggregation Network (PANet) [23], which enhances the localization capability by propagating strong responses of low-level features based on the fact that edges or instance parts are strong indicators to accurately localize objects. Finally, the head of YOLOv5 allows for multi-scale predictions. This is relevant for this work since dirty spots can drastically vary in size. This also allows for different camera distances in relation to the floor.

YOLOv5 framework provides different models based on the number of network parameters and GFLOPs (Giga Floating Point Operation Per second). All the models provided by YOLOv5 were studied in this work in order to select the best performing one. The training was done on the generated dataset and the testing was done on the ACIN dataset. 90% of the generated dataset was used for training and the remaining 10% for validation. In order to tackle overfitting, Transfer Learning [24] was implemented by freezing the backbone, taking advantage of the pre-trained weights on the COCO dataset. The optimizer used was the Stochastic Gradient Descent (SGD) [25] with a learning rate of 0.01 and a decay of 0.0005. The image size was set to 640 × 640, the batch size was set to 16, and the models were trained for 50 epochs, saving the best weights. The training was done with a Nvidia GeForce RTX 3080 GPU and an AMD Ryzen 5 5600X 6-Core 3.7 GHz CPU. Four different experiments were conducted. In the first experiment, all the models provided by the YOLOv5 framework were trained on the synthetic dataset for two classes: solid dirt and liquid dirt. In the second experiment, floors that contain objects were removed from the synthetic dataset which was then used to train a YOLOv5m6 model. This was done to observe the impact of using some floor images with objects on the models' performance. In the third experiment, a new dataset without liquid dirt was generated to train a YOLOv5m6 model. This was done to observe the impact of dirt variety on the models' performance. Finally, in the fourth experiment, a dataset using the same floor images but with random noise for the dirty spots was generated to train a YOLOv5m6 model. These spots have the same shape as the dirt samples used to generate the prior datasets, however they were filled with random solid colors to simulate noise. This was done to demonstrate that generating a synthetic dataset consisting of floors with specific patterns and a series of random spots that break that regularity is not sufficient to train a network capable of detecting dirty spots more accurately than our proposal.

3. Results

In this Section, the experimental results are shown for an Intersection over Union (IoU) ≥ 0.5. This Section is divided into four Subsections addressing the four experiments mentioned in Section 2.

3.1. Multi-Class: Solid Dirt and Liquid Dirt

As mentioned in Section 2, the different models provided by the YOLOv5 framework were trained on the generated dataset in order to select the best performing one in this application. Those models are defined by the network size. Hence, the smallest network is YOLOv5s6, the medium network is YOLOv5m6, the large network is YOLOv5l6, and the largest network is YOLOv5x6. Table 1 provides a better insight about these models.

Table 1. Parameters and FLOPs of each model.

	YOLOv5s6	YOLOv5m6	YOLOv5l6	YOLOv5x6
Parameters (Millions)	16.8	35.7	76.8	140.7
GFLOPs	12.6	50.0	111.4	209.8

Figure 8 illustrates the precision-recall curves on the ACIN dataset for solid dirt and liquid dirt.

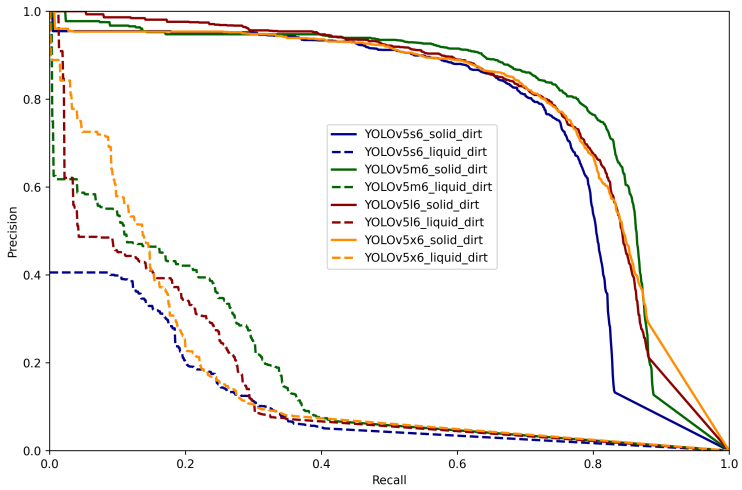

Figure 8. Precision-recall curves on the ACIN dataset for solid dirt (solid lines) and liquid dirt (dashed lines).

Table 2 shows the results on the ACIN dataset for two classes: solid dirt and liquid dirt. Table 3 shows the mAP obtained for each class. Table 4 shows the results for binary classification (dirty or not dirty), which can be relevant in situations where the floor-cleaning robots do not need to distinguish between different types of dirtiness. The time per inference in each model is shown in Table 5 and a comparison with existing methods that were used by [3,4] described in Section 1 is shown in Table 6.

Table 2. Testing on the ACIN dataset for two classes: solid dirt and liquid dirt.

	YOLOv5s6	YOLOv5m6	YOLOv5l6	YOLOv5x6
mAP	0.427	**0.488**	0.472	0.475
Precision	0.532	0.546	0.525	0.564
Recall	0.457	0.521	0.495	0.465
F1	0.492	0.533	0.510	0.510

Table 3. mAP obtained for each class.

	YOLOv5s6	YOLOv5m6	YOLOv5l6	YOLOv5x6
Solid dirt	0.743	0.799	0.789	0.784
Liquid dirt	0.112	0.177	0.155	0.167

Table 4. Testing on the ACIN dataset for binary classification: Dirty or not dirty.

	YOLOv5s6	YOLOv5m6	YOLOv5l6	YOLOv5x6
mAP	0.657	**0.733**	0.720	0.724
Precision	0.795	0.806	0.778	0.804
Recall	0.618	0.703	0.664	0.662
F1	0.695	0.751	0.716	0.726

Table 5. Time per inference.

	YOLOv5s6	YOLOv5m6	YOLOv5l6	YOLOv5x6
Time (ms)	**2.3**	5.7	8.4	18.6

Table 6. Comparison with existing methods on the ACIN dataset for two classes: solid dirt and liquid dirt.

	MobileNet-SSD	YOLOv3	Our Proposal
mAP	0.057	0.381	**0.488**

From Tables 2 and 4 it is possible to observe that YOLOv5m6 attains the best mAP in both multi-class classification and binary classification. Therefore, YOLOv5m6 was selected for the following experiments. Figure 9 shows the confusion matrix of YOLOv5m6 tested on the ACIN dataset. False positives reflect the number of instances that dirt was incorrectly detected in a spot that is clean, and false negatives reflect the number of instances that dirty spots were not detected.

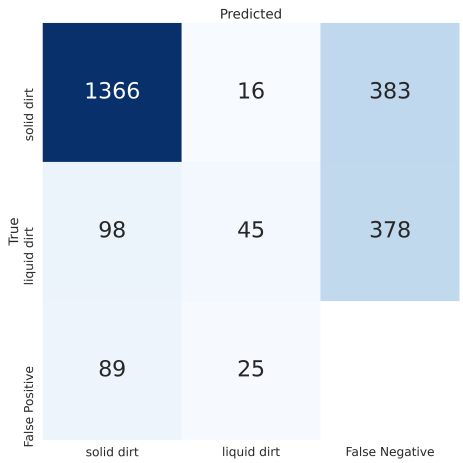

Figure 9. Confusion matrix of YOLOv5m6 on the ACIN dataset.

Figure 10 shows the precision-recall curves for validation and test of YOLOv5m6.

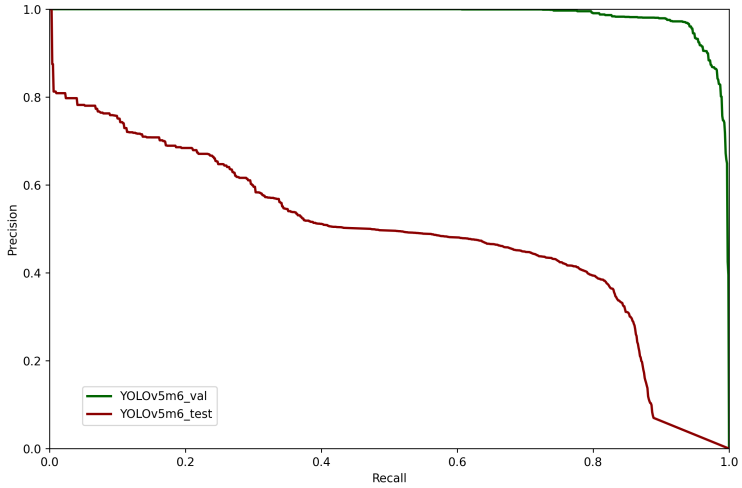

Figure 10. Precision-recall curves for validation and test of YOLOv5m6 when trained for two classes: solid dirt and liquid dirt.

Figure 11 shows some accurate inferences made by YOLOv5m6 and Figure 12 shows some inaccurate inferences made by YOLOv5m6.

Figure 11. Accurate inferences made by YOLOv5m6.

Figure 12. Inaccurate inferences made by YOLOv5m6.

3.2. Removing Floor Images with Objects

As mentioned in Section 2, in this experiment a new YOLOv5m6 model was trained on the synthetic dataset using only floor images that do not contain objects. Table 7 shows the results of this model on the ACIN dataset.

Table 7. Testing on the ACIN dataset with a YOLOv5m6 model trained on the synthetic dataset using only floor images that do not contain objects.

	mAP	Precision	Recall	F1
YOLOv5m6	0.405	0.557	0.444	0.494

Figure 13 shows the confusion matrix of this model when tested on the ACIN dataset.

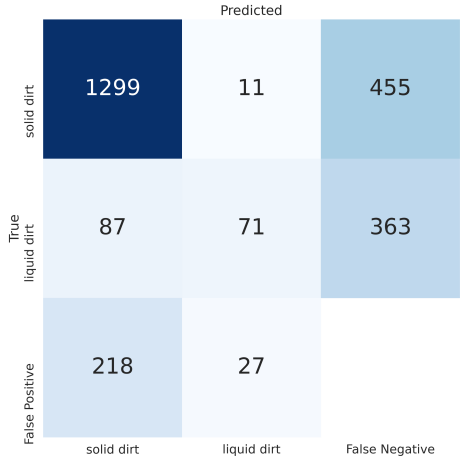

Figure 13. Confusion matrix of YOLOv5m6 on the ACIN dataset when trained using only floor images that do not contain objects.

3.3. Removing Liquid Dirt Samples

As mentioned in Section 2, a new dataset without liquid dirt was generated and used to train a new YOLOv5m6 model. Table 8 shows the results of this trained YOLOv5m6 model on the ACIN dataset for one class: solid dirt.

Table 8. Testing on the ACIN dataset for one class: solid dirt.

	mAP	**Precision**	**Recall**	**F1**
YOLOv5m6	0.874	0.905	0.822	0.862

Figure 14 shows the confusion matrix of this model when tested on the ACIN dataset.

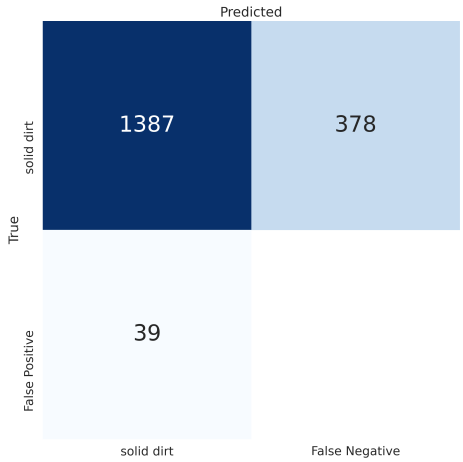

Figure 14. Confusion matrix of YOLOv5m6 on the ACIN dataset when trained for one class: solid dirt.

Figure 15 shows the precision-recall curves for validation and test of YOLOv5m6 when trained for one class: solid dirt.

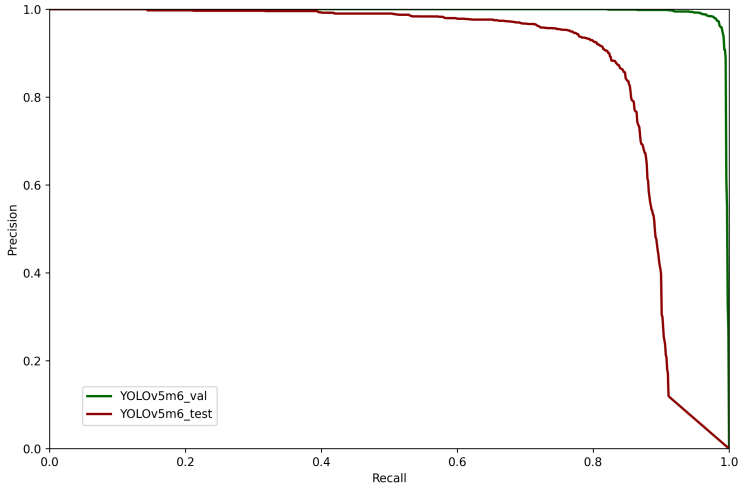

Figure 15. Precision-recall curves for validation and test of YOLOv5m6 when trained for one class: solid dirt.

3.4. Generating a New Dataset with Random Noise

As mentioned in Section 2, a new dataset with random noise was generated and used to train a new YOLOv5m6 model. Figure 16 shows some samples of this dataset.

Figure 16. Samples from the generated dataset with random noise.

Table 9 shows the results of this trained YOLOv5m6 model on the ACIN dataset for one class: solid dirt.

Table 9. Testing on the ACIN dataset for one class: solid dirt.

	mAP	Precision	Recall	F1
YOLOv5m6	0.584	0.803	0.512	0.625

4. Discussion

In this Section, the experimental results shown in Section 3 are discussed. In Figure 8 the precision-recall curves of all models are shown. This essentially indicates that when there is a high area under the curve (high recall and high precision), the model has a low false positive rate and a low false negative rate, that is to say the model is accurate. The area under the curve is also known as mAP. In these same Figures, it is possible to observe that the curves are quite similar, but YOLOv5m6 attained the best results despite having the second lightest network out of the 4 models, consequently being selected for the remaining experiments. This is reflected on the mAP shown in Tables 2 and 4. Theoretically, if one increases the variety of floors, solid dirt and liquid dirt in the generated dataset used for training, the results can be improved. In Table 6 a comparison with the base technologies from works [3,4] is made. These models were trained with the generated dataset and tested on the ACIN dataset. MobileNet-SSD took significantly more training time than YOLOv3 and YOLOv5, and attained the worst results. YOLOv3 converged much faster

than MobileNet-SSD, however it did not perform as well as our proposal on real data. This is expected because YOLOv5 architecture is a direct improvement over YOLOv3.

The generated dataset did not include any floors, solid dirt and liquid dirt of the ACIN dataset that was used for testing. This shows that generating a synthetic dataset to train a YOLOv5 model to detect dirty spots on the floor is a legitimate way to tackle the lack of data in this application. This is specially true if the cleaning robot does not need to distinguish between solid dirt and liquid dirt, as can be seen in Table 4. YOLOv5m6 attained a mAP of 0.733 for binary classification, which is a 50.2% increase from the best result in a multi-class environment shown in Table 2. The results could have been more accurate if the ACIN dataset was not so precisely annotated. For instance, dirt that was still far away from the camera was annotated, dirt that was partially visible was annotated, dirt that is hardly visible to the human eye because of lighting conditions was annotated, and so on. All of these annotations contributed for the false negatives shown in Figure 9. The worst results shown in Table 2 can be explained through Figures 8 and 9, and Table 3. As it is possible to observe in Figure 8, the models struggle to recognize liquid dirt (dashed curves). Figure 9 shows more details about this problem: Only 143 out of 521 liquid dirt samples were detected (27.4%), however out of those 143 liquid dirt samples, only 45 were correctly classified as liquid dirt, while 98 were classified as solid dirt. This can be observed in the left image of Figure 12: While the model was able to detect liquid dirt, the classification was wrong. However, this is expected because the number of liquid dirt samples used for the dataset generation was only 15, since no liquid dataset was found. There is not enough variety in liquids for the models to generalize. In contrast, the number of solid dirt samples used during the dataset generation was 141 thanks to the Trashnet dataset [19], which allowed the model to generalize and be accurate when facing new solid dirt samples.

In order to observe what was the impact of including floor images that contain objects, a new YOLOv5m6 model was trained on the synthetic dataset without this type of floor images, following the same training procedure described in Section 2. Then this model was tested on the ACIN dataset and the results can be seen in Table 7 and Figure 13. Since those floor images that contain objects were removed from the training dataset, this model struggled to distinguish between dirty spots and objects. The mAP decreased from 0.488 to 0.405, which was a 17% decrease. The reason behind this is detailed in Figure 13. As expected, the number of false positives was significantly higher (144.9%) and there were 7.5% more false negatives when compared to the confusion matrix in Figure 9, which covers the results of a YOLOv5m6 model trained with the original generated dataset. This shows that our proposal on using floor images that contain objects during the dataset generation has a positive impact on the results, mainly because it reduces the amount of false positives.

In order to observe the impact of dirt variety, a new dataset without liquid dirt was generated. A new YOLOv5m6 model was trained on this dataset following the same training procedure described in Section 2. This model was then tested on the ACIN dataset and attained a mAP of 0.874, as shown in Table 8. This improvement can be better visualized when comparing Figure 10 with Figure 15. It is also possible to observe in the confusion matrix shown in Figure 14 that removing the liquid samples from the training dataset improved the model when tasked with detecting only solid dirt. The number of true positives increased, and the number of false negatives and false positives decreased when compared to Figure 9. These results further show the negative impact of having a small variety of liquid samples in the synthetic dataset.

In order to observe the impact of using real dirt samples instead of random noise, a new dataset with spots that have the same shape of the dirt samples but with a random solid color was generated. These spots strive to simulate noise and to break the regularity of the floor pattern. A new YOLOv5m6 model was trained on this dataset following the same training procedure described in Section 2. This model was then tested on the ACIN dataset and attained a mAP of 0.584, as shown in Table 9. This was a 33.2% decrease from the results observed in Table 8. This decrease shows that using real dirt samples in the

dataset generation leads to better results than merely adding random noisy spots to break the regularity of the floor pattern.

A problem that the YOLOv5 models faced was the presence of objects that appear partially in the image, such as chairs, wires, shoes, and so on. Since they appear partially in the image, the YOLOv5 models struggle to recognize if they are a dirty spot or not. This problem can be seen in the center and right images of Figure 12. One fast solution is programming the cleaning robot in such a way that it ignores inferences on the borders of the image. This would significantly reduce the number of false positives.

As can be concluded in this Section, there is still some work to be done in this area. The problem of false positives on the borders of the image can be a complex one. Ideally one wants the model to distinguish if there is a dirty spot in the image even if this spot is partially visible. The hypotheses for future work is that by adding more complex floor images with partially visible objects that are not dirty spots for the dataset generation can help tackling this problem. This was already done to some extent in this work as can be seen in Figure 5. However, the criteria on choosing these floor images was simply adding objects that the cleaning robot could face in a real-world environment, and not the specific problem that is being discussed. Another problem that was found during this work is a type of dirtiness that is not being discussed in the literature: Stains. Stains on the floor can be as relevant as solid dirt and liquid dirt. The ACIN dataset has a great deal of stains on its floors which were sometimes detected as dirty spots, however since they were not annotated these inferences were considered as false positives. Figure 17 illustrates this problem.

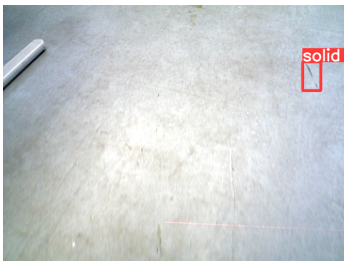

Figure 17. Stain detected by YOLOv5.

As can be observed in Figure 17, the solid dirt inference is considered a false positive, however this region represents a stain. For future work, all the stains present on the ACIN dataset will be annotated. A new dataset will be generated considering this new type of dirtiness. All of this will be made public as well. Training on hybrid datasets, by combining labeled data with synthetic data, or using semi-supervised learning techniques may outperform the methods described in this paper and will be considered for future work.

As a summary for future work:

- The number of liquid dirt samples will be increased for generating the new dataset.
- Floor samples with partially visible objects that are not dirty spots will be added.
- Stains on the ACIN dataset will be annotated.
- Stain samples will be added for generating the new dataset.

5. Conclusions

In this work, the YOLOv5 framework was explored for detecting dirty spots on the floor. Due to the lack of data, a tool found in the literature was upgraded and adapted to generate a dataset that had a rich variety on floors, solid dirt and liquid dirt. This dataset was used to train YOLOv5 models. The ACIN dataset was used for testing these models. However, the ACIN dataset was partially annotated and therefore our work included further annotating this dataset, tripling the number of labels. The results were quite satisfactory considering that no floor patterns, solid dirt or liquid dirt from the ACIN dataset were used for generating the training dataset. This shows that our approach

is able to generalize relatively well since the training dataset is significantly different from the testing dataset. This is specially true for the experiment where only solid dirt was considered. YOLOv5m6 attained the best results on both multi-class and binary classification, with a mAP of 0.488 and 0.733 respectively. It was also demonstrated the positive impact of using floor images that contain objects for generating a synthetic training dataset. This approach helps the models to distinguish between dirty spots and objects, which significantly reduces the number of false positives. It was also concluded in this work that using real dirt samples to generate the synthetic dataset leads to better results than using random noisy spots.

Some problems were found during this work and will be tackled in the future. For instance, the YOLOv5 models struggled to detect the liquid dirt because there was a lack of variety on this front during the dataset generation. By generating a new dataset without liquid dirt and using it to train a YOLOv5m6 model, the mAP obtained when testing with the ACIN dataset was 0.874, which further shows the impact of dirt variety in the synthetic dataset. These models also struggled to distinguish partially visible objects from dirty spots. Lastly, a new type of dirtiness (stains) that is not being discussed in the literature was found to be relevant.

Author Contributions: Conceptualization, D.C., A.J.R.N., P.G. and P.F.; methodology, D.C., A.J.R.N. and P.G.; software, D.C.; validation, D.C., A.J.R.N. and P.G.; formal analysis, D.C., A.J.R.N. and P.G.; investigation, D.C.; resources, P.F.; data curation, D.C.; writing—original draft preparation, D.C.; writing—review and editing, D.C., A.J.R.N., P.F. and P.G.; visualization, D.C.; supervision, A.J.R.N., P.G. and P.F.; project administration, A.J.R.N., P.G. and P.F.; funding acquisition, i-RoCS: Research and Development of an Intelligent Robotic Cleaning System (Ref. POCI-01-0247-FEDER-039947). All authors have read and agreed to the published version of the manuscript.

Funding: This work was developed in the scope of the project "i-RoCS: Research and Development of an Intelligent Robotic Cleaning System" (Ref. POCI-01-0247-FEDER-039947), co-financed by COMPETE 2020 and Regional Operational Program Lisboa 2020, through Portugal 2020 and FEDER.

Data Availability Statement: Data generation tool: https://github.com/ddcanedo/synthesis_tool (accessed on 30 November 2021); Generated dataset: https://uapt33090-my.sharepoint.com/:f:/g/personal/danielduartecanedo_ua_pt/EhzEtCBQIGFKkjshLSGNc-YBacasS0rGmR5zqmFkTfo0rA?e=E6LnKd (accessed on 30 November 2021); ACIN dataset new annotations: https://uapt33090-my.sharepoint.com/:f:/g/personal/danielduartecanedo_ua_pt/ErCqPZVvoOBCu4TrgbCjxwcBwJ_SwElLh9qAgPDaL4BFCQ (accessed on 30 November 2021).

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

Abbreviations

The following abbreviations are used in this manuscript:

ACIN	Automation and Control Institute
mAP	Mean average precision
GMM	Gaussian Mixture Model
SSD	Single-Shot MultiBox Detector
CNN	Convolutional Neural Network
SVM	Support Vector Machine
YOLO	You Only Look Once
RCNN	Region Based Convolutional Neural Network
CSPNet	Cross Stage Partial Network
PANet	Path Aggregation Network
GFLOPs	Giga Floating Point Operation Per second
SGD	Stochastic Gradient Descent
IoU	Intersection over Union
ms	milliseconds

References

1. Kang, M.C.; Kim, K.S.; Noh, D.K.; Han, J.W.; Ko, S.J. A robust obstacle detection method for robotic vacuum cleaners. *IEEE Trans. Consum. Electron.* **2014**, *60*, 587–595. [CrossRef]
2. Grünauer, A.; Halmetschlager-Funek, G.; Prankl, J.; Vincze, M. The power of GMMs: Unsupervised dirt spot detection for industrial floor cleaning robots. In *Towards Autonomous Robotic Systems*; Gao, Y., Fallah, S., Jin, Y., Lekakou, C., Eds.; Springer: Cham, Switzerland; Guildford, UK, 2017; pp. 436–449. [CrossRef]
3. Ramalingam, B.; Lakshmanan, A.K.; Ilyas, M.; Le, A.V.; Elara, M.R. Cascaded machine-learning technique for debris classification in floor-cleaning robot application. *Appl. Sci.* **2018**, *8*, 2649. [CrossRef]
4. Bormann, R.; Wang, X.; Xu, J.; Schmidt, J. DirtNet: Visual Dirt Detection for Autonomous Cleaning Robots. In Proceedings of the 2020 IEEE International Conference on Robotics and Automation (ICRA), Paris, France, 31 May–31 August 2020; pp. 1977–1983. [CrossRef]
5. Duda, R.O.; Hart, P.E. *Pattern Classification and Scene Analysis*; Wiley: New York, NY, USA, 1973; Volume 3, pp. 9–15.
6. Connolly, C.; Fleiss, T. A study of efficiency and accuracy in the transformation from RGB to CIELAB color space. *IEEE Trans. Image Process.* **1997**, *6*, 1046–1048. [CrossRef] [PubMed]
7. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. SSD: Single Shot MultiBox Detector. In *Computer Vision—ECCV 2016*; Leibe, B., Matas, J., Sebe, N., Welling, M., Eds.; Springer: Cham, Switzerland; Amsterdam, The Netherlands, 2016; pp. 21–37. [CrossRef]
8. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *arXiv* **2017**, arXiv:1704.04861.
9. Liu, W.; Wang, Z.; Liu, X.; Zeng, N.; Liu, Y.; Alsaadi, F.E. A survey of deep neural network architectures and their applications. *Neurocomputing* **2017**, *234*, 11–26. [CrossRef]
10. Noble, W.S. What is a support vector machine? *Nat. Biotechnol.* **2006**, *24*, 1565–1567. [CrossRef] [PubMed]
11. Perez, L.; Wang, J. The effectiveness of data augmentation in image classification using deep learning. *arXiv* **2017**, arXiv:1712.04621.
12. IPA Dirt Detection. Available online: http://wiki.ros.org/ipa_dirt_detection (accessed on 2 September 2021).
13. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
14. Girshick, R. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
15. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft coco: Common objects in context. In *Computer Vision—ECCV 2014*; Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T., Eds.; Springer: Cham, Switzerland; Zurich, Switzerland, 2014; pp. 740–755. [CrossRef]
16. Jocher, G. ultralytics/yolov5: V5.0—YOLOv5-P6 1280 models, AWS, Supervisely and YouTube integrations. *Zenodo* **2021**. [CrossRef]
17. ACIN Dataset. Available online: https://goo.gl/6UCBpR (accessed on 18 November 2021).
18. Hawkins, D.M. The problem of overfitting. *J. Chem. Inf. Comput. Sci.* **2004**, *44*, 1–12. [CrossRef] [PubMed]
19. Yang, M.; Thung, G. *Classification of Trash for Recyclability Status*; CS229 Project Report; Stanford University: San Francisco, CA, USA, 2016.
20. Marcu, A.; Licaret, V.; Costea, D.; Leordeanu, M. Semantics through Time: Semi-supervised Segmentation of Aerial Videos with Iterative Label Propagation. In Proceedings of the Asian Conference on Computer Vision, Kyoto, Japan, 30 November–4 December 2020.
21. Tzutalin. LabelImg. Available online: https://github.com/tzutalin/labelImg (accessed on 6 September 2021).
22. Wang, C.Y.; Liao, H.Y.M.; Wu, Y.H.; Chen, P.Y.; Hsieh, J.W.; Yeh, I.H. CSPNet: A new backbone that can enhance learning capability of CNN. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Seattle, WA, USA, 14–19 June 2020; pp. 390–391.
23. Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. Path aggregation network for instance segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 8759–8768.
24. Pan, S.J.; Yang, Q. A survey on transfer learning. *IEEE Trans. Knowl. Data Eng.* **2009**, *22*, 1345–1359. [CrossRef]
25. Bottou, L. Large-scale machine learning with stochastic gradient descent. In Proceedings of the COMPSTAT'2010, Paris, France, 22–27 August 2010; Lechevallier, Y., Saporta, G., Eds.; Physica-Verlag HD: Paris, France, 2010; pp. 177–186. [CrossRef]