A quantitative comparison of automated cleaning techniques for web scraped image data of 'Smart Cities'

de Witte, B.; Bouarfa, S.; Geradts, Z.

DOI
10.1145/3529446.3529457

Publication date
2022

Document Version
Final published version

Published in
IPMV 2022

License
Article 25fa Dutch Copyright Act (https://www.openaccess.nl/en/in-the-netherlands/you-share-we-take-care)

Link to publication

Citation for published version (APA):

General rights
It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations
If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: https://uba.uva.nl/en/contact, or a letter to: Library of the University of Amsterdam, Secretariat, Singel 426, 1012 WP Amsterdam, The Netherlands. You will be contacted as soon as possible.

UvA-DARE is a service provided by the library of the University of Amsterdam (https://dare.uva.nl)
A quantitative comparison of automated cleaning techniques for web scraped image data of ‘Smart Cities’

Bob de Witte  
boh.de.witte@hotmail.com  
University of Amsterdam  
Amsterdam, Netherlands

Soufiane Bouarfa  
soufiane.bouarfa@accenture.com  
Accenture Data & AI  
Amsterdam, Netherlands

Zeno Geradts  
Z.J.M.H.Geradts@uva.nl  
Netherlands Forensic Institute, University of Amsterdam  
Den Haag, Netherlands

ABSTRACT
This paper implements and compares four automated image cleaning techniques through the ResNet-34 Convolutional Neural Network, motivated by the need to reduce manual cleaning efforts of large image datasets. For each of these techniques, the relation with the literature on automated image cleaning is identified. Each of the four techniques uses a specific criterion to identify and remove unwanted images from datasets. The criteria range from identifying images with text, through identifying images with a specific size or tonal distribution, up to identifying images with a specific training loss value. In order to evaluate the four cleaning techniques, ResNet-34 was trained with web scraped images corresponding to 15 object classes of ‘Smart Cities’, and accuracy results were obtained through testing on the CalTech 256 dataset subset. The results show that manual cleaning outperforms automated cleaning techniques on all four criteria. However, analysis reveals that the individual automated techniques or a combination thereof can initially be deployed on large datasets before manual verification to reduce workload and increase dataset stability.

CCS CONCEPTS
- Computing methodologies → Modeling, Machine Learning;  
- Human-centered computing → Human computer interaction;  
- Information systems → Information retrieval.

KEYWORDS
automated cleaning techniques, web images, image classification, smart cities

1 INTRODUCTION
In recent years, the interest in computer vision has evolved rapidly. Computer vision aims to see, identify and understand the visual world as humans do, which outperforms the human brain in specific tasks [12]. This rapid evolution can be mainly ascribed to the increasing number of innovative techniques to perform a wide range of Artificial Intelligence (AI) tasks. In addition, the availability of Graphics Processing Units (GPUs) in the cloud makes it easier to execute large computational tasks. This is particularly beneficial for areas such as image classification [19, 28] and object detection [38, 46] where Deep Learning (DL) plays a central role. Deep Learning makes use of complex networks, such as Convolutional Neural Networks (CNNs). The technique of training CNNs, often used for computer vision applications [18], has shown its success in different sectors like healthcare [11, 13], crime [10], self-driving cars [2] and forensics [3], to name a few.

The performance of DL models highly correlates with the quality and volume of the training data. In the field of computer vision, training data includes relevant images or videos. Consequently, several popular image datasets have emerged in the field ranging from smaller datasets such as the MNIST [29] and the CalTech 256 [17] dataset to more extensive datasets such as Google Open Images [15] and ImageNet [8]. However, these well-known datasets contain limited object classes and often require significant human efforts to be developed. This is typically done through collecting a large number of images from the web. The internet has opened a vast collection of visual content which can be accessed relatively easily to prepare tailored image datasets, e.g. through scraping images. For instance, Kaggle and Google Dataset Search are two sources to fulfill this task. Also, search engines like Google, Yahoo or Bing provide an almost infinite resource of visual content. The object class of collected images mainly depends on the use case. For instance, images can represent examples of dangerous situations [10, 16] or aircraft defects [9], which can be used to train DL models. The data collection process typical involves cleaning raw data from noise and bias and removing mislabeled images [7] to have a better performing DL algorithm. This usually requires significant efforts by the researchers. This has led the research community to develop various automated image cleaning techniques to reduce manual efforts and increase datasets quality. These cleaning techniques (CTs) operate in a broad spectrum of complexity, ease of implementation and improvement in accuracy results.

This paper investigates various automated image cleaning techniques, motivated by the need to reduce manual cleaning efforts of large datasets. To achieve this, a novel literature review is first
conducted to give an overview of existing automated image cleaning techniques. These techniques usually aim to have clear and good quality dataset where the object class is dominantly present. Then, a systemic evaluation of selected cleaning techniques is performed through the ResNet-34 CNN. Accuracy results are compared to manual cleaning performance to assess to what extent manual cleaning can be avoided.

The paper is organized as follows. Section 2 gives an overview of automated cleaning techniques and complementary background on the image classification problem. Section 3 describes the methodology used. Section 4 summarizes the results and analysis. Section 5 discusses the results, and Section 6 provides the conclusion.

2 RELATED WORK

This paper focuses on the data preparation of an image classification problem. Data preparation includes 1) data collection, 2) data cleaning and 3) data augmentation [20]. Because of the focus of the paper, data augmentation is not used in the data preparation.

2.1 Data collection

Visual data for image classification problems can be obtained via various methods. Depending on the computer vision problem, not all state-of-art datasets can be applied to all computer vision problems, because these problems are too specific. Therefore, tailored datasets are collected in other ways. The two main ways to collect more visual data are data synthesis and data searching [20]. Data synthesis means generating data to mimic the real world, including techniques like Generative Adversarial Networks (GANs). Data searching means looking for more data in a broad sense, for instance by utilising the web.

As the internet has almost an infinite volume of all types of data, using the web is a common way to enrich or create datasets. For instance, Google Street View images were used to detect and map traffic signs [6] or to characterise food cultivation along roadside transects [39]. Another example is the use of Google Earth satellite images to evaluate the earth’s well-being, for example [25, 41]. Several other web browsers, such as Yahoo, Flickr and Bing, also provide search engines for images. These web-browsers even have open-source libraries for scraping images automatically, or an API can be enabled to access them. For the purpose of this research, no additional permission requirements need to be made to use the web data.

The web is a very intuitive source for collecting images. For instance, ImageNet [8] also uses search engines for data collection. Two main problems often occur when comparing a web scraped dataset to a better known, existing dataset:

(1) The used keyword may not exactly match the search results
(2) The web images might be mislabeled or might include several unwanted labels

These problems typically arise because search engines usually operate in a high-precision low-recall regime and tend to be biased towards pictures with a single, centred object with a clean background [7]. This leads to the practice of cleaning the dataset.

2.2 Data cleaning

Data cleaning is the process of detecting and correcting/removing incorrect, inaccurate or irrelevant parts of a dataset. An image dataset must contain enough relevant and quality images to build a reliable DL model. Therefore, a certain fraction of the dataset is usually removed to improve the image dataset, which includes crowd-sourcing, automated techniques or engaging experts.

However, this paper focuses on automated cleaning techniques to improve an image dataset. Several automated cleaning techniques, which have shown improvement in previous research, are presented in Table 1 shortly. Various cleaning techniques stated in this table are used in combination with different datasets and models, so quantifiable results cannot be compared to each other or the outcomes of this paper.

2.2.1 Automated cleaning techniques. Roughly, automated cleaning techniques can be divided into two categories: 1) Techniques that use complex architectures/methods to identify mislabeled images and outliers in the entire dataset, 2) Techniques that look at a particular aspect within every individual image. Table 1 represents a concise overview of automated cleaning techniques.

The first category includes C Ts 1-6 of Table 1. All techniques use multiple steps to identify possible outliers using various methods (e.g. Deep Learning, Reinforcement Learning). These techniques are not focused on one criterion in every image, but aim to identify outliers within an entire dataset.

The second category includes C Ts 7-12 of Table 1. All these techniques focus on one characteristic within images and are often used to clean a dataset manually. These techniques target a particular criterion to identify bad images individually, whereas the first category tries to identify bad images in an entire dataset.

2.2.2 Manual cleaning. Manual cleaning is often the most reliable in improving model performance, as bad images can be detected more easily. Different guidelines and criteria can be set for this cleaning technique, depending on what an image classifier must be trained on. Some criteria can be subjective, which can make manual cleaning less consistent. Also, cleaning an image dataset is a monotone process for the cleaner and can thus result in less consistency within an entire dataset.

2.2.3 Hybrid cleaning. Hybrid cleaning combines both manual and automatic cleaning, which can be used to leverage the advantages of machines and human expertise. For instance, in [14] a clustering algorithm was used to find relevant images in an image dataset interactively.

3 METHODOLOGY

The research consisted of three phases: 1) Gathering a training, validation and test set, 2) Running experiments on datasets using a fixed CNN and 3) Evaluating experiments.

3.1 Data

The data consisted of a training, validation and test dataset. First, in total 9222 images were scraped using the Bing image downloader.

1https://github.com/ostrolucky/Bulk-Bing-Image-downloader/
Table 1: Overview of automated image cleaning techniques.

<table>
<thead>
<tr>
<th>Cleaning Technique (CT)</th>
<th>Description</th>
<th>Execution efforts</th>
<th>Used dataset(s)</th>
<th>References</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Image cleaning using transfer learning</td>
<td>This CT consists of two steps: 1) Transfer learning is used to create a better classifier, 2) Minority classes are cleaned up with a determined threshold value.</td>
<td>Medium</td>
<td>667 fish classes, 168972 images</td>
<td>[31]</td>
</tr>
<tr>
<td>2. Image cleaning using the loss function</td>
<td>This CT consists of two phases: 1) Training an initial model to identify training images with the highest loss. 2) Remove images with the highest loss.</td>
<td>Medium</td>
<td>N.A.</td>
<td>Cleaning tool of [21]</td>
</tr>
<tr>
<td>3. Image cleaning using reinforcement learning</td>
<td>This CT uses reinforcement learning to determine the optimal preprocessing steps of image data automatically.</td>
<td>High</td>
<td>MNIST, SVHN, CIFAR-10, DOGCAT [23, 27, 29, 36]</td>
<td>[33]</td>
</tr>
<tr>
<td>4. ‘Well Begun Is Half Done’</td>
<td>This CT generates high-quality seeds to grow an image dataset subsequently. It uses clustering for similar images and detects outliers that might not be the same object class.</td>
<td>High</td>
<td>Web-23 [30]</td>
<td>[42]</td>
</tr>
<tr>
<td>5. ‘ImageDC’</td>
<td>This CT uses Deep Learning to 1) Clean out images from the minority class and to 2) Adopt the low recognition rate to remove noise labeled images.</td>
<td>High</td>
<td>3 Twitter datasets of 427, 765, 1121 classes</td>
<td>[43]</td>
</tr>
<tr>
<td>6. Image cleaning using model consensus, explainability and confident learning</td>
<td>This CT uses model consensus, explainability and confident learning to identify bad (quality), mislabeled, unclear images per category.</td>
<td>High</td>
<td>ImageNet [8]</td>
<td>[24]</td>
</tr>
<tr>
<td>7. Detecting images with text</td>
<td>This CT uses ‘Optical Character Recognition’ (OCR) for identifying text in images.</td>
<td>Medium</td>
<td>Flowers 500+ classes, 150 images per class</td>
<td>practical: [35]</td>
</tr>
<tr>
<td>8. Removing low-quality images</td>
<td>This CT removes low-quality images from the image dataset.</td>
<td>Low</td>
<td>Set5, Set14, B100, Urban100 [5, 22, 32, 44]</td>
<td>[26]</td>
</tr>
<tr>
<td>9. Identifying multiple objects in images</td>
<td>This CT tries to find multiple objects per image to estimate whether an image is cluttered.</td>
<td>Medium</td>
<td>N.A.</td>
<td>e.g. [38]</td>
</tr>
<tr>
<td>10. Noise reduction within images</td>
<td>This CT aims to lower the noise in an image which often arises because of additional unnecessary pixels causing the loss of information.</td>
<td>Low</td>
<td>BSDS500 [1]</td>
<td>[34], practical: [4]</td>
</tr>
<tr>
<td>11. Removing duplicate images</td>
<td>This CT removes duplicate images in a dataset.</td>
<td>Low</td>
<td>N.A.</td>
<td>e.g. [45]</td>
</tr>
<tr>
<td>12. Detecting tonal distribution in images</td>
<td>This CT uses individual pixels in an image and identifies unique characteristics of every pixel. This way, tonal distribution can be detected to classify an image as artificial (including cartoons and drawings).</td>
<td>Low</td>
<td>Flowers 500+ classes, 150 images per class</td>
<td>e.g. [8, 17], practical: [35]</td>
</tr>
</tbody>
</table>

This amount of images was the maximum amount of images available in the Bing server for the 15 used object classes at the time of scraping and was, therefore, the limit of the Bing server. In total 15 different object classes were defined, which could potentially be used in a use case on ‘Smart Cities’ (identifying objects or trash on streets): bathtub, bulldozer, car tire, covered wagon, fire hydrant, fire truck, gas pump, hot tub, mattress, picnic table, refrigerator, school bus, telephone box, traffic light and washing machine. All BMP, PHP, GIF, AXD and ASP file types were removed from the scraped dataset. To keep a balanced dataset for all object classes, 500 images per class were selected randomly, resulting in 7500 images in total. A total of 1500 images (20%; 100 per object class) was used as a validation set and was kept constant for all experiments. The other 6000 images (80%; 400 per object class) were used as the training set. The training dataset was used as input before every cleaning technique. Thus, after every cleaning technique, the total amount of training images was less than 6000.

The used test set is a subset of the CalTech 256 dataset. This dataset originally contains 30607 images from 256 object classes. However, the same 15 object classes as the training set were used, resulting in 1729 test images. Again, the number of images from the object class with the least amount of images was applied to all object classes to keep object classes balanced. This resulted in 84 test images for every class, which were randomly selected; 1260 test images in total. The dataset was kept separately from the training set and was used to test how well models performed.

3.2 Modeling

3.2.1 Baseline modeling. The Fast.ai high-level API was used for modeling [21]. Fast.ai is built on top of PyTorch [37]. The training
set from Section 3.1 was the input for modeling. After that, the validation set validated the model, and the test set evaluated the model on the ground truth. Default parameter values form Fast.ai were set for building the Datablock. All images were resized to a size of 224x224 pixels, which is needed to parse the image to the CNN. No data augmentation was done, as model improvement for various cleaning techniques was the main focus, rather than absolute model accuracy. Data augmentation would have only added more variables, making it harder to evaluate cleaning techniques individually.

The ResNet-34 architecture was used without the use of transfer learning. ResNet-34 is a well-known, stable CNN and is relatively good at generalising validation data because it is not extremely complex compared to other ResNet architectures [19]. In general, more complex models are more sensitive for overfitting, take longer to train and use more memory space. Although transfer learning generally results in better accuracy results, it was not used because of two reasons: 1) Some object classes overlap with object classes from ImageNet. This way, the model would be biased towards certain classes, which was not preferred. 2) Transfer learning could affect the impact of a cleaning technique, as results might not change significantly because the model is still trained properly.

In total, 30 epochs were executed for training almost 22 million model parameters. The number of epochs was based on flattening accuracy scores of the first tests. The cross-entropy loss function was used to penalise deviations in model predictions and actual labels.

Manual cleaning. Manual cleaning was done as a separate experiment. Manual cleaning (sometimes combined with automated techniques) is still the best way to clean an image dataset. This way, the results of cleaning techniques could be compared to the optimal way of cleaning an image dataset. Cleaning guidelines of [17] were followed to decide what images to remove.

3.2.2 Experiments of automated cleaning techniques. Besides the baseline and the manual cleaning experiments, four other experiments were executed. Each experiment had its cleaning technique. The cleaning techniques were selected so that the expected result was relevant for the web scraped dataset and that it was achievable to perform multiple experiments in the given time frame. All experiments were executed with the same modeling parameters and settings as the baseline (Section 3.2.1).

The cleaning techniques ImageDC, Well Begun Is Half Done, Image cleaning using reinforcement learning, and Image cleaning using model consensus, explainability and confident learning were considered ‘high’ implementation effort and were therefore not used. Image cleaning using transfer learning was excluded because no transfer learning was used during modeling. Removing duplicate images was taken care of in the scraping phase. Multiple different objects per image and noise reduction within an image were considered not relevant for the scraped dataset based on the first impression of the training image dataset. For these reasons four CTs were investigated:

- CT 1: Detecting images with text
- CT 2: Detecting tonal distribution
- CT 3: Image cleaning using the loss function
- CT 4: Removing low-quality images

The numbering of the CTs has no correlation with the numbering of CTs in Table 1, but indicates the chronological order of execution of experiments during the study. For CT 2, 3 and 4 a certain threshold was involved resulting in the number of images that was cleaned out. The method of Equal Error Rate (EER) is often used to evaluate thresholds. However, the method involves too many experiments per CT to apply in this research. Therefore, four different thresholds per CT were tested.

CT 1: Detecting images with text. CT 1 aims to identify images that contain text. ‘Optical Character Recognition’ (OCR) is a tool where text in images is extracted to be evaluated. The open-source library Tesseract [40] was used for this. Images where only white spaces were detected were not removed. Regular expressions (RegEx) and the Enchant library were used to identify valid English words. Images containing (a) valid word(s) with more than three characters were removed. This resulted in a smaller training set of 5367 images (10.6% of the images were removed).

CT 2: Detecting tonal distribution. All hue and lightness values were extracted per image. The same hue and lightness value between pixels is a measure of similarity. All unique combinations of hue-lightness pairs were summed in every image. The 1000 most frequent hue-lightness pairs per image were summed, and the fraction of occurrence of these pairs was calculated by dividing this summation by the total number of pixels in each image. This resulted in a score of what fraction these 1000 pairs contained in the image. Artificial images have fewer unique pairs than natural images, so the fraction which the 1000 most frequent pairs include is usually higher for artificial images. A distribution of fraction values over the images is presented in Figure 1. The images with the 200, 400, 600 and 800 highest fractions were subsequently taken out before individual experiments.

CT 3: Image cleaning using the loss function. This CT uses the baseline model from Section 3.2.1. This way, the training images with the highest training loss can be detected. The higher the loss, the farther off the prediction is from the actual label according to the used model and loss function. These images are thus confusing for the algorithm, and it will therefore overcompensate for these
A quantitative comparison of automated cleaning techniques for web scraped image data of ‘Smart Cities’

IPMV 2022, March 25–27, 2022, Hong Kong, China

images. Subsequently, 200, 400, 600 and 800 training images with the highest loss were taken out before modeling. The cross-entropy loss function per image is given by

$$-\frac{1}{N} \sum_{n=1}^{N} y_n \log p_n = -\log p_t,$$

where $N$ is the number of classes, $y_n$ the true label (0 or 1), $p_n$ the prediction probability of every class and $p_t$ is the prediction probability of the true label. As only one object class per image is present, $y_n$ is zero except if the value of the label matches the label of the image. Figure 2 shows the distribution of cross-entropy loss values for all images.

CT 4: Removing low-quality images. This technique investigates the size of every image by calculating $\sqrt{\text{width} \cdot \text{height}}$. Figure 3 shows the distribution of the sizes of the training images. Consequently, the smallest 200, 400, 600 and 800 images were taken out for modeling.

3.3 Evaluation

For quantitative evaluation the accuracy metric was used. The accuracy metric is a percentual score of correctly predicted images for all object classes. For both the validation and test set the accuracy scores on object class level were calculated. Another measurement of how well a CT performs was used. The overlap between images that were removed for every experiment and manually cleaned images was compared. The better the overlap, the better the expectation of the performance of a CT, as manually removed images were considered bad images.

Besides quantitative analysis, a qualitative analysis was done to evaluate whether the technique removed the right images. This way, mistakes in the cleaning technique itself could be revealed and possible explanations for quantitative results might be identified. Overall remarkable insights were evaluated per techniques on an object class level.

4 RESULTS AND ANALYSIS

Figure 4 presents accuracy scores on the validation and test set for each experiment. As expected, all models with fewer training images than the manual cleaned model (experiments where 800 images were removed) perform worse than the manual model on the test set. Results on the test set are considered leading, as this was the ground truth CalTech 256 dataset subset. Sub-experiments per CT are distinguished with the number of removed images per sub-experiment.

Figure 5 presents the fraction of correctly removed images per experiment compared to the total number of removed training images.

Baseline. The baseline model was trained to have a comparable model for the results of other cleaning techniques. The model scores a 77.5% and 71.4% accuracy on the validation and test set respectively.

Manual. Manual cleaning, as expected, resulted in the best accuracy score on the test set with an improvement of 0.9%. However, a one-tailed Z-test for two proportions between the baseline and manually cleaned model results in a p-value of 0.312, which is therefore not a significant result by assuming a significance level of 0.05. As the manually cleaned model performed best, other scores logically are not significant either.

It appears that not all object classes have the same quality of web scraped images, as the number of removed images per object class varies from 11 (class: ‘picnic table’) to 95 (class: ‘traffic light’). Thus, based on used cleaning criteria, the quality of web scraped images varies per object class. However, this has nothing to do with absolute accuracy scores on object class level, as every object has its own difficulty level to train. Therefore, some classes need more training images than others. Looking at the accuracy score on the validation set, which slightly improved with 0.2%, manual cleaning did not stand out compared to other experiments of cleaning techniques. CT2_200 and CT2_400 score better on the validation set.

Correctlyremoved images. Figure 5 shows a remarkable trend considering the total number of removed training images compared to the fraction of correctly removed images. Correctly removed images means the number of images that were removed by both
Figure 4: Accuracy scores for all experiments on the validation set (left) and the CalTech 256 test set (right). Scores of the baseline model, manual model and different CT models are separated by colour. Sub-experiments for CT 2, CT 3 and CT 4, where 200, 400, 600 and 800 images were removed, are distinguished within belonging colours.

Figure 5: Fraction of correctly removed images per cleaning technique compared to the total number of removed images for all experiments. The higher the fraction of good removals, the more images were removed correctly. Manual cleaning is considered optimal, so the fraction would be 1. Random cleaning would have a fraction value of $\frac{768}{6000} = 0.128$, as for every removed image the chance is 0.128 to be a good removal for the used dataset.

5 DISCUSSION

On a high level, the value of the used cleaning techniques is not significant for the model environment of this paper. Even manual cleaning does not give a significant improvement on accuracy scores. This indicates that the outcome has no proper correlation with the way the training images were cleaned. As cleaning an image dataset manually is a common practice for image classification, possible explanations could more likely lie in the used dataset/use case, model decisions or other decisions within the experimental setup. To give an example of how the experimental setup could make a big difference, in [31] a significant result was achieved after removing 396 minority classes out of 667 object classes. This way, the raw dataset was modified before baseline modeling, which is a different choice regarding the experimental setup. Therefore, outcomes of this paper can not be compared to outcomes of original reference papers, like [31].

Besides, it seems too complicated to reveal why certain experiments get these accuracy score results; these results are based on an estimation of millions of parameters of the ResNet-34 architecture. Every training image is responsible for every parameter adjustment, making it almost impossible to pinpoint how these accuracy scores were obtained. Different modeling decisions probably do not give significant results either, as the absolute accuracy scores would change (model optimisation) rather than the improvement in accuracy scores for different experiments.

However, cleaning an image dataset leads typically to a more reliable and stable model, as bad training images do not influence negative changes in model parameters. Removing bad images does not particularly lead to significant accuracy score improvements for the used experimental setup. The fundamental question here for the cleaning criteria is what problem do you want to solve with your trained model? To answer this question, a training image dataset must be created that resembles test images from the ‘ground truth’ test set, as the model will only be as good as the quality of the training data. This could also be the reason that manual cleaning does not result in the best scores on the validation set, as the training set was not cleaned on criteria of the validation set. Also, regarding the cleaning technique and the manual cleaning. All cleaning techniques, excluding CT 1 because it had only one experiment, remove the highest fraction of bad images when the least amount is removed (200 images). Removing more images leads to a decrease when looking at good removals as a fraction of the total removals. This might indicate that cleaning techniques perform better when looking at the ‘worst’ cases per technique. CT 2 and CT 3 consistently show the best scores, as they remove the worst images proportionally for all experiments.

CT 4 constantly performs the worst. Interestingly, however, CT2_200 has the highest accuracy scores on both validation and test set of all experiments.
the used test set, even though the CalTech 256 dataset is still used in research, a newer, more extensive test set could be sought.

In this paper manual cleaning criteria were followed from the original paper of the used test set and was thus considered optimal. For automating a cleaning technique to fulfill these cleaning criteria, the technique must identify several criteria within images. Automating this could lead to more consistency, but possibly making it less accurate as well. Figure 5 shows that the maximum overlap is removed images between manual cleaning and automated cleaning is only around 35% (CT2_200). This strengthens the aforementioned, as an automated cleaning technique can probably not account for all cleaning criteria.

Every cleaning technique aims for specific criteria and aspects in images that are considered bad for training a model. Manual cleaning takes advantage of the ability of a human brain to identify multiple criteria for an image. Automated cleaning is not able to do that. With the research approach of this paper, no conclusion can be drawn on how well an automated cleaning technique performance where it was meant for, as it was not documented why an image was removed when cleaning manually. This could be potential future research to investigate what combination of automated cleaning techniques could potentially cover multiple cleaning criteria adequately.

In general, removing data at random is bad for a model, as more data often is better and results in a more consistent model. Regarding the number of cleaned out images per CT, no tangible value for the threshold could be determined which was optimal, as results were deviating too much between different threshold values. However, this study indicates that it is essential to be careful when removing images with an automated cleaning technique, as too many good images would be removed otherwise.

Ultimately, it is not worth using automated cleaning techniques for small datasets, as cleaning small datasets manually is not time-consuming. Even for the dataset used in this paper, it took around two hours to clean it manually, which is not a major human effort. A balance between model accuracy improvement and time advantage must be made for more extensive datasets. For models that need a fast PoC it could be helpful to use automated cleaning.

6 CONCLUSION

This paper implements and compares four automated image cleaning techniques through the ResNet-34 CNN. Though accuracy variations exist between the techniques, none of the evaluated techniques outperform manual cleaning. However, the results indicate that combining multiple cleaning techniques in a hybrid setup can significantly reduce manual efforts and increase dataset stability (see Figure 5). Researchers can first deploy cleaning techniques to automatically remove unwanted images based on different criteria, before manual verification. This human-machine collaboration could lead to significant time savings as cleaning the entire dataset manually would not be required. This is especially relevant for large datasets such as ImageNet [15] and Google Open Images [8], as smaller datasets can be cleaned manually much easily. This work can be further extended by investigating additional cleaning techniques from Table 1, either on the same considered dataset or other datasets. Another line of research could focus on how to best incorporate automated cleaning techniques into the data preparation process which is manually driven. This would be especially relevant for Proof of Concept use cases, as DL models usually aim to have the highest performance, which is normally achieved with manual cleaning as indicated by the results of this research.

REFERENCES


Advances in neural information processing systems, 32, 8026-8037.


