



Fine-Grained WEEE Product Retrieval Using Hybrid Search and Re-ranking

Ajibola Obayemi¹(✉)  and Khuong An Nguyen² 

¹ University of Brighton, East Sussex, Brighton BN2 4GJ, UK
e.a.obayemi@brighton.ac.uk

² Royal Holloway University of London, Surrey, Egham TW20 0EX, UK
khuong.nguyen@rhul.ac.uk

Abstract. This paper addresses the challenge of efficient and accurate retrieval of Waste Electrical and Electronic Equipment (WEEE) products, specifically printer cartridges. Traditional recycling methods, heavily reliant on manual inspections, are labour-intensive and inefficient. To overcome these limitations, we propose a hybrid search and re-ranking approach which incorporates a combination of NLP-based methods. By leveraging these methods, we aim to improve the recall@1k performance of WEEE product retrieval, facilitating more efficient and sustainable WEEE recycling practices. The experimental results achieve a recall@1k score of 60.18%, showing the effectiveness of our approach. This improvement has significant implications for the recycling industry, enabling more accurate identification and sorting of WEEE products and ultimately contributing to a more sustainable circular economy.

Keywords: Product retrieval · Hybrid search · Vector database · Vector search · Semantic search · Keywords based search · BM25 · Reranking · WEEE

1 Introduction

Fine-grained product retrieval is a challenging task, particularly when faced with products that exhibit intra-class variability (differences occurring in the same product category) and inter-class similarity (resemblance between different product categories). These factors often lead to misclassification errors in traditional retrieval systems. By addressing this problem, we hope to reduce the frequency of these errors, which can be difficult to overcome using conventional methods [1–4].

Hybrid search and re-ranking are NLP techniques employed in fine-grained product retrieval. These techniques can address the challenge of retrieving relevant products from extensive databases, often containing thousands or millions of entries with subtle differences in attributes and descriptions. By combining multiple NLP methods, such as keyword and semantic-based search, hybrid search can efficiently process user queries or keywords to identify the most pertinent

product suggestions. Re-ranking further refines these results, ensuring that the most relevant products are presented at the top. This approach is well-suited for tasks like WEEE product retrieval and automation, where accurate and efficient product retrieval is essential [25–27, 30, 31].

WEEE products, specifically printer ink cartridges [8], are widely used in schools, businesses, and events, yet recycling rates are low. Most end up in landfills, where they slowly decompose and release harmful chemicals. Sorting these cartridges for reuse and recycling is a key challenge and solving this challenge can help create a more sustainable environment by reducing e-waste [16, 18, 23, 32]. This paper presents a hybrid search and re-ranking approach to enhance WEEE product retrieval using the TAIMD-17k dataset, which comprises 17,000 image-text pairs across 31 distinct product classes (available online at <https://github.com/multimodal-research/taimd-17k>) [8].

Our key contributions include: (1) An introduction of a hybrid search method for fine-grained retrieval of WEEE products, (2) A re-ranking algorithm to optimise retrieval efficiency and accuracy and (3) A demonstration of the practical application and effectiveness of this hybrid search and re-ranking framework within an industrial setting.

2 Related Works

[18] developed an automated system for recognising WEEE (specifically digital cameras) by using OCR to extract manufacturer information from labels. This information was then matched against a pre-existing database via template matching. However, their method requires manual template creation and incurs a high computational cost for improved accuracy. For text similarity, they employed the basic Levenshtein distance, suitable for their simple dataset. In contrast, our work employs a more sophisticated approach that involves hybrid search and a re-ranking score, as detailed later.

Similarly, [32] created a system to recognise text on battery images, aiming to automate the sorting of battery packs and pouch cells from WEEE. Their three-stage approach involved extracting text, filtering it, and matching it against a database of battery specifications. They experimented with various matching algorithms, settling on partial ratio, which functions by comparing substrings, a fundamentally different approach from the one we present.

3 Methods

This section examines fine-grained product retrieval through natural language processing (NLP), with a specific focus on optical character recognition (OCR)-based text extraction from product labels, such as printer cartridges. It highlights how information retrieval methods overcome noisy OCR output to accurately match this text with a product database and associate it with predefined class labels. The subsequent sections will elaborate on each stage of this process.

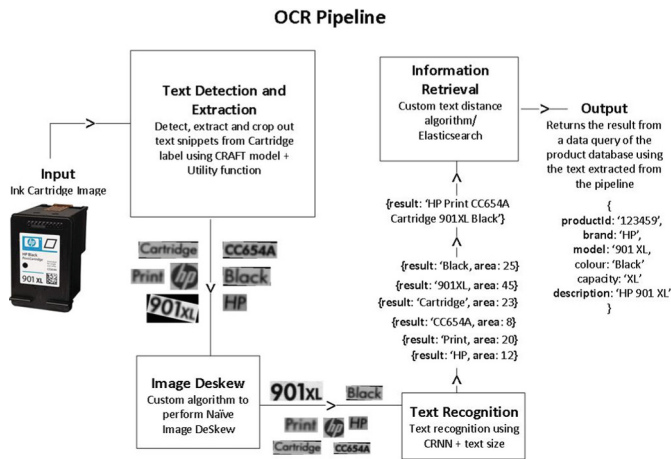


Fig. 1. From Image to Text: A Visual Guide to the OCR Pipeline.

3.1 Text Detection, Extraction and Cropping

1. CRAFT & EAST models for text detection: Several works have addressed text detection in natural images [5–7,9]. We compared EAST [10] and CRAFT [5] text detection models, followed by Tesseract v4 [11–13] text recognition, to determine the most effective approach for our application. CRAFT outperformed EAST in accurately localising text, especially with complex layouts and challenging conditions (see results in Table 1).
2. Text Extraction and Cropping: We describe a five-step methodology for precisely extracting detected text regions from images, a crucial process for document analysis and OCR. It begins with a utility function taking CRAFT model coordinates and the original image. A mask is then created based on these coordinates, and a polygon defines the text region (ROI), with the area outside being removed. The resulting masked region, including padding, is cropped and then passed to a rotation and skew correction algorithm. This method aims to isolate text while preserving context, as illustrated in Fig. 2.

Table 1. An evaluation of the preliminary experiments carried out to choose the best text detection and extraction method for our dataset.

Method	Brand - HP	Brand - Epson
EAST Text Detection	0.5%	9%
CRAFT	67%	47%

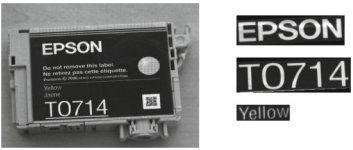


Fig. 2. Detected text with different crop sizes

3.2 Image Rotation and Naïve Skew Correction

Preliminary testing revealed the necessity of skew correction and rotation normalisation to achieve reliable text recognition. To address text-level rotation, we implemented a fundamental heuristic that analysed the image’s aspect ratio. We used Gaussian blur, thresholding, and contour detection (OpenCV) for horizontal images to estimate rotation from the innermost contour and correct the image. Vertical images were rotated in 90-degree increments until a horizontal orientation was achieved. Corrected images were then processed by a CRNN-based STR framework (Fig. 3).



Fig. 3. Illustrative examples of image correction results obtained through applying the rotation and skew correction algorithm. The algorithm improves the blurry images for OCR.

3.3 CRNN Based Text Recognition

The Clova AI four-stage STR framework [5] served as the foundation for our text recognition method. This framework builds upon the successful integration of Recurrent Neural Networks (RNNs) and Convolutional Neural Networks (CNNs) for STR, as demonstrated in prior works by [14, 15, 17]. By effectively integrating these neural network architectures, the STR framework excels at recognising text within complex and challenging image scenarios.

3.4 Product Information Retrieval with Elasticsearch (BM25)

We aim to classify images into predefined product classes using human-readable labels by integrating an end-to-end OCR pipeline (Fig. 1) with a robust information retrieval system. The OCR pipeline, as shown, needs to link visual text

to product categories, a task that requires information retrieval. To overcome the noise in CRNN-generated OCR, we use Elasticsearch [24], a scalable and flexible text search engine based on Apache Lucene [10, 19]. Its ability to handle large text volumes and support various ranking algorithms makes it suitable; we specifically use the BM25 algorithm. This algorithm calculates relevance using term frequency, inverse document frequency, and document length. Elasticsearch's BM25 implementation scores document relevance by considering the frequency and importance of search terms. Term Frequency is the count of a keyword in a document, and IDF represents its inverse frequency across the dataset [20, 28, 29]. The mathematical formula for BM25 is given in Eq. 3.4.

$$score(D, Q) = \sum_{i=1}^n IDF(q_i) \cdot \frac{f(q_i, D) \cdot (k_1 + 1)}{f(q_i, D) + k_1 \cdot (1 - b + b \cdot \frac{|D|}{avgdl})}$$

score(D, Q) is the relevance score of document D given a query Q.

$\sum_{i=1}^n$ is expressed as the summation across all query terms.

$IDF(q_i)$ is expressed as the Inverse document frequency of query term i.

$f(q_i, D)$ is the Frequency of query term i in document D.

k_1 & b are Tuning parameters.

$|D|$ represents the Length of document D.

$avgdl$ is the Average document length in the collection.

3.5 Product Information Retrieval with Vector Search

Text-based product retrieval assigns products to categories based on textual descriptions. Although traditional methods used linguistic rules and keyword search, recent advancements show that vector-based representations improve performance [21, 22]. Vector search (Fig. 5) represents data as vectors, efficiently retrieving information from large datasets. Compared to traditional keyword-based search methods, vector search offers several advantages, as it handles semantic similarity, is scalable, and is crucial for information retrieval. As data grows in volume and complexity, vector search is positioned to play a critical role in shaping the future of information retrieval and search technologies [21, 22].

3.6 Product Retrieval with Hybrid Search

This section introduces a hybrid search framework, merging the advantages of both keyword-based and vector-based (semantic) search methods. Our approach involves a three-stage process: (1) Initial retrieval: BM25 retrieves initial results based on keyword matching, (2) Vector database query: Pass the query into a vector database and retrieve similar vectors with their similarity scores. (3) Re-ranking: Introduce a re-ranking algorithm that combines the scores from keyword and vector-based methods to identify the most relevant documents. This hybrid methodology addresses the limitations of traditional search approaches by providing a more comprehensive and accurate search experience.

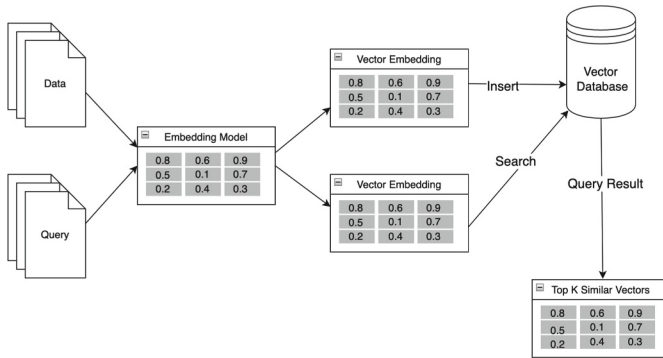


Fig. 4. Vector search systems convert data into meaningful vector embeddings and store them in a vector database for retrieval.

4 Experiments

The experiments detailed in this section empirically validate our proposed methods (described in Sects. 3.4, 3.5, and 3.6).

4.1 Elasticsearch (with BM25)

We implemented a text extraction and information retrieval system to leverage the OCR pipeline outlined in Fig. 1. Text detection was performed using the CRAFT model [5], which uses a TPS-ResNet-BiLSTM-Attn architecture. The detected text regions were cropped from the input image and processed by the CRNN model [14, 15, 17] for text recognition. CRNN’s sequence-to-sequence architecture, coupled with an attention mechanism, enabled accurate transcription of the extracted text. The recognised text was then concatenated into a unified string and utilised as a query for information retrieval in Elasticsearch. Our cartridge data, indexed in Elasticsearch with the BM25 similarity metric, was efficiently searched using a multi-match query. This approach allowed the retrieval of the model IDs, product IDs, colours, and capacities based on the extracted text content. We present results from this experiment in Table 2.

4.2 Vector Search

We implement Qdrant [25], an open-source vector database, as our vector database for experiments in this section. It provides a RESTful API, compatible with multiple programming languages, and supports exact and approximate nearest-neighbour searches. Qdrant supports various data types, search criteria, and JSON payloads for filtering. To begin, we create a new collection within the Qdrant database. This collection is designed to store and organise our data, and we configure it to use COSINE similarity for calculating distances between vectors. Subsequently, the model data is uploaded, followed by creating a function

and API endpoint. This facilitates the submission of queries and the retrieval of associated scores.

4.3 Hybrid Search

In this section, we discuss our experiment, which combines Elasticsearch, a powerful full-text search engine, with BM25, a widely recognised algorithm for ranking search results by computing the relevance score of a given query. We have integrated vector search techniques to augment our search capabilities further, allowing us to efficiently compare and retrieve similar items based on their semantic representations. Finally, we have implemented an additional layer specifically designed to facilitate the selection of the closest options and re-ranking search results, enabling us to refine the order of returned items based on various factors such as contextual information and domain-specific knowledge.

In this experiment, the OCR-derived query from the image label was submitted to both the Elasticsearch system described in Sect. 4.1 and the vector search system detailed in Sect. 4.2. The resulting outputs were evaluated using a custom re-ranking algorithm (see the steps highlighted in algorithm 4.3) that compares the original text to the returned results.

- 1: **Input:** strings x , y
- 2: **Output:** similarity score
- 3: *Step 1, convert text to lowercase*
- 4: *Step 2, calculate compression similarity of x & y*
- 5: *Step 3, tokenise x*
- 6: *Step 4, tokenise y*
- 7: *Step 5, create a set of x tokens*
- 8: *Step 6, create a set of y tokens*
- 9: *Step 7, create a union of x & y tokens*
- 10: *Step 8, create a vector*
- 11: *Step 9, calculate the cosine similarity of x & y vectors*
- 12: *Step 10, calculate the compression similarity of x & y vectors*
- 13: **Return:** $\max(\text{cosine} + \text{compression similarity})$

The formula for calculating compression similarity is given by Eq. 1 and Cosine similarity is represented in Eq. 2 below:

$$\text{compression_similarity_score} = \frac{\text{length}(\text{compress}(\text{concatenate}(\text{text}_a, \text{text}_b)))}{\text{length}(\text{compress}(\text{text}_a)) + \text{length}(\text{compress}(\text{text}_b))} \quad (1)$$

$$\text{cosine_similarity} = \frac{c}{(\sum_{i=1}^n l_1[i] \cdot \sum_{j=1}^m l_2[j])^{0.5}} \quad (2)$$

5 Results

This section compares the performance of three search methods for fine-grained WEEE product retrieval: traditional keyword search (Elasticsearch/BM25),

semantic search (vector similarity), and our novel hybrid search incorporating re-ranking. By analysing the results, we aim to highlight the benefits and limitations of each approach and demonstrate the potential of our hybrid method.

As shown in Table 2, we observe that to achieve a higher initial recall, our hybrid search strategically combines exact keyword matching with a nuanced understanding of semantic similarity, ensuring that we capture both directly relevant and conceptually related WEEE items. Subsequently, our re-ranking algorithm refines these initial results by filtering and prioritising to elevate the most relevant and accurate matches.

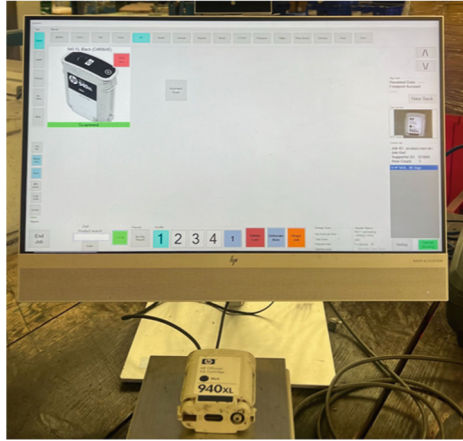


Fig. 5. This figure illustrates a real-world application of our hybrid search system, demonstrating the retrieval of WEEE products through its integrated components and re-ranking algorithm within an industrial setting.

6 Discussion and Future Work

Our work introduces a hybrid search and re-ranking algorithm that enhances the retrieval of fine-grained WEEE products based on textual information. We considered keyword, semantic, and hybrid search methods for WEEE product classification. Our experiments evaluated the proposed methods using the TAIMD-17k text and image dataset. The results, summarised in Table 2, demonstrate the enhanced performance of the hybrid search method (with our re-ranking algorithm), achieving an overall recall@1k of 0.6018. Further analysis revealed that while BM25 was more tolerant of noisy text, vector search struggled in such conditions. Our re-ranking algorithm, combined with the hybrid search, improved search results, increasing accuracy from 0.5947 to 0.6018 and specifically increased the class-level performance of difficult classes.

Table 2. Class level results comparing all three methods (BM25, Vector search and Hybrid search) using the Recall@1k evaluation metric.

Class	BM25	Vector Search	Hybrid Search	Total per class
100144 Black Standard HP-301 CH561E	177	173	175	238
100426 Cyan XL HP 933XL CN054A	50	33	50	57
100426 Magenta XL HP 933XL CN054A	85	24	85	87
100426 Yellow XL HP 933XL CN054A	93	43	93	94
106465 Tri-Colour Standard HP 301 CH562E	159	162	134	238
153537 Cyan Standard Canon 526	34	21	60	65
153537 Magenta Standard Canon 526	76	77	77	93
153537 Yellow Standard Canon 526	46	36	74	80
153593 Black Standard Canon 551	5	6	0	30
153593 Cyan Standard Canon 551	1	1	1	62
153593 Grey Standard Canon 551	0	0	0	5
153593 Magenta Standard Canon 551	5	5	4	70
153593 Yellow Standard Canon 551	3	3	3	70
155608 Cyan XL HP 951XL CN046A	86	50	86	92
155608 Magenta XL HP 951XL CN046A	75	10	75	78
155608 Yellow XL HP 951XL CN046A	61	27	61	68
159433 Black Standard Epson 18	0	0	0	56
159433 Cyan Standard Epson 18	0	24	0	71
159433 Magenta Standard Epson 18	0	17	0	56
159433 Yellow Standard Epson 18	0	3	0	57
166188 Black Standard Epson T1281	81	4	81	82
166188 Cyan Standard Epson T1282	52	44	52	54
166188 Magenta Standard Epson T1283	60	15	60	60
166188 Yellow Standard Epson T1284	40	13	40	42
166195 Black Standard Epson T0711	0	0	0	92
166195 Cyan Standard Epson T0712	71	23	72	73
166195 Magenta Standard Epson T0713	66	20	66	67
166195 Yellow Standard Epson T0714	6	0	6	6
166209 Cyan XL Canon 551XL	30	31	25	98
166209 Magenta XL Canon 551XL	44	18	43	90
166209 Yellow XL Canon 551XL	10	9	10	50
Class Accuracy	0.5947	0.3746	0.6018	

Future work will investigate the potential of integrating large language models (LLMs) with retrieval-augmented generation (RAG) for fine-grained product retrieval, building on the knowledge gained from this research to improve our results and generalise our methods to other related downstream tasks.

References

1. He, X., Peng, Y.: Fine-grained visual-textual representation learning. IEEE Trans. Circuits Syst. Video Technol. **30**(2), 520–531 (2019)

2. Mafra, A., et al.: Fine-grained image classification and retrieval by combining visual and locally pooled textual features. In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision* (2020)
3. Xuan, Q., et al.: Evolving convolutional neural network and its application in fine-grained visual categorisation. *IEEE Access* **6**, 31110–31116 (2018)
4. Baz, I., Yoruk, E., Cetin, M.: Context-aware hybrid classification system for fine-grained retail product recognition. In: *2016 IEEE 12th Image, Video, and Multi-dimensional Signal Processing Workshop (IVMSP)*, 1–5 (2016)
5. Baek, J., et al.: What is wrong with scene text recognition model comparisons? Dataset and model analysis. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2019)
6. Borisyuk, F., Gordo, A., Sivakumar, V.: Rosetta: large scale system for text detection and recognition in images. In: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (2018)
7. Liao, M., Shi, B., Bai, X.: Textboxes++: a single-shot oriented scene text detector. *IEEE Trans. Image Process.* **27**(8), 3676–3690 (2018)
8. Obayemi, A., Nguyen, K.A.: Advanced data mining and applications. In: Sheng, Q.Z., et al. (eds.), vol. 15392. Springer Nature Singapore (2025). <https://doi.org/10.1007/978-981-96-0850-8>
9. Qin, S., et al.: Towards unconstrained end-to-end text spotting. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2019)
10. Zhou, X., et al.: EAST: an efficient and accurate scene text detector. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2017)
11. Smith, R.: An overview of the Tesseract OCR engine. In: *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*, vol. 2. IEEE (2007)
12. Patel, C., Patel, A., Patel, D.: Optical character recognition by open source OCR tool tesseract: a case study. *Int. J. Comput. Appl.* **55**(10), 50–56 (2012)
13. Zacharias, E., Martin, T., Bénédicte, B.: Image processing based scene-text detection and recognition with tesseract. *arXiv preprint: arXiv:2004.08079* (2020)
14. Cheng, Z., et al.: Focusing attention: towards accurate text recognition in natural images. In: *Proceedings of the IEEE International Conference on Computer Vision* (2017)
15. Lee, C.-Y., Simon, O.: Recursive recurrent nets with attention modeling for OCR in the wild. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2016)
16. Shahidzadeh, M.H., et al.: Discovering the secret behind managing WEEE: deep learning method in the industry 4.0. *Ann. Oper. Res.*, 1–36 (2023)
17. Shi, B., Bai, X., Yao, C.: An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **39**(11), 2298–2304 (2016)
18. Hayashi, N., Koyanaka, S., Oki, T.: Constructing an automatic object-recognition algorithm using labeling information for efficient recycling of WEEE. *Waste Manage.* **88**, 337–346 (2019)
19. Zhang, Y., Li, J.: Research and improvement of search engine based on Lucene. In: *2009 International Conference on Intelligent Human-Machine Systems and Cybernetics*, vol. 2. IEEE (2009)
20. Kim, M.-Y., et al.: Legal information retrieval and entailment based on bm25, transformer and semantic thesaurus methods. *Rev. Socionetw. Strat.* **16**(1), 157–174 (2022)

21. Van Gysel, C., de Rijke, M., Kanoulas, E.: Learning latent vector spaces for product search. In: *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management* (2016)
22. Xian, J., et al.: Vector search with OpenAI embeddings: Lucene is all you need. In: *Proceedings of the 17th ACM International Conference on Web Search and Data Mining* (2024)
23. Pan, X., Wong, C.W., Li, C.: Circular economy practices in the waste electrical and electronic equipment (WEEE) industry: a systematic review and future research agendas. *J. Clean. Prod.* **365**, 132671 (2022)
24. Gormley, C., Tong, Z.: *Elasticsearch: The Definitive Guide: A Distributed Real-Time Search and Analytics Engine*. O'Reilly Media, Inc. (2015)
25. Singh, P. N., Talasila, S., Banakar, S.V.: Analysing embedding models for embedding vectors in vector databases. In: *2023 IEEE International Conference on ICT in Business Industry & Government (ICTBIG)*. IEEE (2023)
26. Taipalus, T.: Vector database management systems: fundamental concepts, use-cases, and current challenges. *Cogn. Syst. Res.* **85**, 101216 (2024)
27. Bassani, E., Pasi, G.: A multi-representation re-ranking model for personalized product search. *Inf. Fusion* **81**, 240–249 (2022)
28. Wu, H.C., et al.: Interpreting TF-IDF term weights as making relevance decisions. *ACM Trans. Inf. Syst. (TOIS)* **26**(3), 1–37 (2008)
29. Kim, S.-W., Gil, J.-M.: Research paper classification systems based on TF-IDF and LDA schemes. *HCIS* **9**(1), 1–21 (2019). <https://doi.org/10.1186/s13673-019-0192-7>
30. Shcherbakov, V., Krasnov, F.: Multi-word term embeddings improve lexical product retrieval. arXiv preprint: [arXiv:2406.01233](https://arxiv.org/abs/2406.01233) (2024)
31. Li, S., et al.: Query rewriting in TaoBao search. In: *Proceedings of the 31st ACM International Conference on Information & Knowledge Management* (2022)
32. Sterkens, W., et al.: Computer vision and optical character recognition for the classification of batteries from WEEE. *Procedia CIRP* **105**, 110–115 (2022)