Proof of concept for a new battery sorting method based on deep learning image classification

Fridolin Blum[1,2], Nils Wieczorek[1], Peer Stelldinger[2]

Abstract: Battery recycling requires efficient sorting based on chemical composition. Traditional methods like X-Ray or Electromagnetic Sensors lack automation, with X-Ray sorting 26 batteries and electromagnetic sorting only 6 batteries per second. We propose using deep learning image classification to detect battery manufacturer and product series. Our prototype includes a conveyor belt, webcam, ring light, and Nvidia Jetson AGX Orin. With a dataset of 9 battery series, we achieved over 99% validation accuracy using a pretrained MobileNetV2 model. The model can classify 50 images per second with limited hardware. This approach offers potential for automated sorting, significantly improving recycling throughput and efficiency. Further research should expand the dataset and explore applicability to other battery types, optimizing the model and hardware configuration.

Keywords: battery recycling; deep learning; image classification

Addresses Sustainable Development Goal 12: Responsible consumption and production

1 Introduction

Battery recycling has gained significant importance due to the number of discarded batteries and their adverse environmental impact. The German battery law, BattG2, mandates all battery distributors to accept the return of batteries, irrespective of their manufacturer, with a minimum requirement of 50% take-back quota. To facilitate efficient recycling, accurate identification of battery chemical composition is essential. In this paper, we focus specifically on mignon batteries (type AA), while other methods such as sewing and rotating disks can be employed for sorting based on shapes and sizes [FM15].

Current state-of-the-art techniques for automatic battery sorting involve weighing plus electromagnetic sensors or X-ray sensors. The weighing plus electromagnetic sensor achieves a sorting performance of 6 batteries per second, while the X-ray sensor achieves 26 batteries per second, both with a sorting accuracy of 98% [FM15]. More recently deep learning based object detection is applied for battery recycling [Ka18, St21]. In this paper, we propose leveraging the success of deep learning in computer vision applications [KSH17, Gi14] and applying deep learning-based image classification to streamline the battery recycling process. Our approach utilizes a convolutional neural network, specifically MobileNetV2, for classifying battery images based on their manufacturer and product series.

[1] Stiftung GRS Batterien, Heidenkampsweg 44, 20097 Hamburg, Deutschland {blum,wieczorek}@stiftung-grs.de
[2] Haw Hamburg, Berliner Tor 7, 20099 Hamburg, Deutschland {peer.stelldinger,fridolin.blum}@haw-hamburg.de

Since companies utilize the same components across product lines, the chemical composition can be obtained by referring to online resources or the International Electrotechnical Commission (IEC) code printed on the battery itself. We opt for image classification instead of optical character recognition (OCR) to take advantage of color and pattern information printed on the battery. It should be noted that manufacturers, product series, and IEC codes are often printed only once per battery, making them not always visible in standard images.

2 Materials and methods

2.1 Hard and software

The neural network training for battery classification was conducted on a laptop with an Nvidia T120 GPU and an Intel I7-11800H CPU. The training script was implemented using the TensorFlow and Keras frameworks version 2.11. For image acquisition, a Logitech C920 HD Pro Webcam was utilized to capture training and live images. The Rollei LUMIS Mini Ring Light Bi-Colour was employed for lighting purposes during image acquisition. Deep learning inference, on the other hand, was performed on the Nvidia Jetson AGX Orin Developer Kit, running Jetson Linux with JatPack 5.1 installed.

2.2 Prototype construction

We constructed a small conveyor belt using toy building blocks and positioned the webcam and ring light at a height of 10cm above it (Fig. 1). The conveyor belt operated at a speed of 10cm/s. To minimize the influence of ambient light, both the light source and the webcam were placed within a plastic box.

All training, validation, and test images were captured using this setup. Additionally, we utilized this setup for our demonstration prototype, in conjunction with the Nvidia Jetson Orin Developer Kit, to classify batteries in a 30fps video stream.

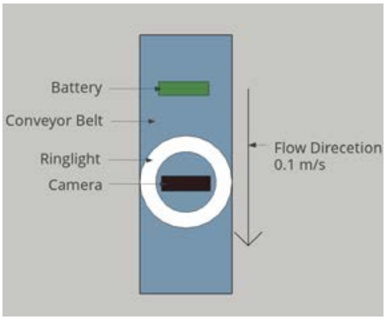


Fig. 1: Building block conveyor belt where the battery classification is made on.

2.3 Dataset

The dataset (Fig. 2) was generated using a webcam, ring light, and the building block conveyor belt (see Section 2.2). A total of 27 different batteries, with 3 batteries per class, were used to create the images. Videos of the batteries passing on the conveyor belt were recorded, and subsequently, each frame from the videos was extracted, resulting in a dataset of 2413 images.

In initial experiments, we achieved training and validation accuracies higher than 99%. However, the battery classification performance on the live video stream was found to be inadequate, which could be attributed to the similarity of the images extracted from every frame of the video (Fig. 3). After the train-validation-split images in the training and validation datasets are nearly identical, therefore overfitting could occur without reducing the validation accuracy.

To address this issue, we extended our dataset by including 931 additional images from new videos. During the extension process, we only extracted every tenth frame to avoid generating highly similar images (Fig. 3 and Fig 4). Additionally, we generated a separate dataset using different batteries, where we also extracted every tenth image. This dataset was reserved solely for testing purposes after completing the training loop, allowing us to simulate new real-world data.

Testing the model trained on the unextended dataset against the test dataset yielded an accuracy of 62%. Train the model with the extended dataset yielded in an accuracy higher 99% (see Section 3) and resulted in an adequate classification performance on the live video stream, with batteries that were not used in any previous dataset.

The final dataset comprises 3344 images distributed across 10 different classes, including 9 distinct battery series and a "no battery"class. For training purposes, the dataset was split into training and validation sets, with 80% of the data allocated for training and 20% for validation. The test dataset consists of 350 images, which is approximately 10% of the combined size of the training and validation datasets.

Fig. 2: Sample images from the dataset. Because the images are made on the running conveyor belt and made with a webcam the images are blurred.

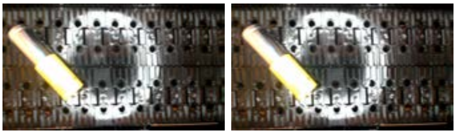


Fig. 3: Similarity of images when using every frame of the video as training image.

Fig. 4: Similarity of images when using only every tenth frame of the video as training image.

2.4 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are deep neural networks used for image and video recognition. They employ multiple convolutional layers to extract visual features at various levels of abstraction, while pooling layers downsample the feature maps to retain vital information. The model's predictions are made using fully connected layers, which map the extracted features to the desired outputs, such as class prediction scores. CNNs have changed image classification by automatically learning filters and features, eliminating the need for handcrafted ones. The typical architecture includes convolutional layers for feature generation and fully connected layers for prediction [GBC16].

2.5 Model selection and adoption for transfer learning

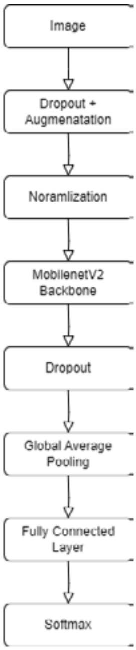


Fig. 5: Model architecture: Pipeline from the input image to the final prediction.

Transfer learning involves initializing a model with weights learned from a different, larger dataset instead of random initialization, resulting in faster training and requiring less data [HBF19]. However, transfer learning may introduce unwanted features from the previous dataset, which can be addressed by fine-tuning the entire network [Ki17].

In this early-stage prototype, only pre-existing models in Keras are considered for the base model, eliminating the need for additional implementation or importing effort. These deep learning models for image classification have been pretrained on the ImageNet database [SVL23]. We used MobileNetV2 [Sa18] despite its lower accuracy on ImageNet compared to ConvNeXtXlarge. Although ConvNeXtXlarge achieves the highest score on ImageNet (86.7%), it is approximately 100 times larger than MobileNetV2, which is the smallest pretrained model.

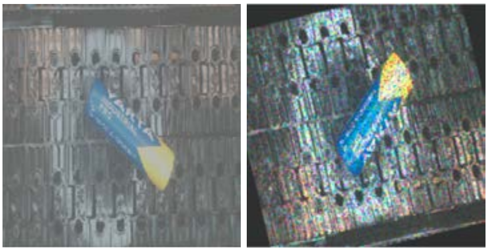
Considering energy consumption as a concern in deep learning [SGM20], we aim to minimize energy usage by selecting a smaller model, contributing to energy-efficient battery sorting design. The choice of a smaller model also leads to faster inference and training times, which is advantageous for our limited hardware (see Section 2.1).

We made slight modifications to the head of MobileNetV2. After the pretrained model backbone, we added a global average pooling layer, a dropout layer with a rate of 0.2, a fully connected layer with 10 output neurons (one for each class), and a SoftMax activation function as the new, untrained model head (Fig. 5). The dropout layer randomly sets 20% of the connections to zero during training to reduce reliance on specific neurons and prevent overfitting. During inference and validation, dropout is disabled. Dropout serves as an ensemble learning approach, where each training step trains a subnetwork, and the results of the subnetworks are combined, leading to more reliable features [Sr14, Hi12].

SoftMax activation is defined by the formula: $\mathrm{SoftMax}(x_i) = \frac{\exp(x_i)}{\sum_j \exp(x_j)}$, where (x_i) is the input vector. SoftMax generates an output vector of the same size as the input vector, with each element ranging from 0 to 1, and the sum of all elements equaling 1. Applying SoftMax to the last layer of the CNN yields an output that represents the probability per class.

Before feeding images into the neural network, we apply image augmentation layers and an image normalization layer (see Section 2.6).

2.6 Image preprocessing



(a) Input Image (b) Augmented Image

Fig. 6: Comparison between the input image (a) and the augmented image with Dropout (b).

To enhance training data variability and prevent overfitting [SK19], we incorporate several image augmentation layers into the model. First, a random flip layer is applied, which flips the image vertically or horizontally. Next, a random rotation layer rotates the image by up to 18 degrees. Additionally, a random contrast and random brightness layer adjust the contrast and brightness by 40%. These data augmentation layers are combined to generate new images from the training dataset (Fig. 6).

To further enhance generalization and prevent the model from relying on individual pixels, a dropout of 0.2 is applied to the input images [Sr14, Hi12]. Following augmentation, all pixel values of the images are scaled down to a range between 0 and 1.

2.7 Transfer learning the Network

The MobileNetV2 model was utilized with pretrained weights from the ImageNet dataset, and its head was replaced with a customized one to suit our requirements (see Section 2.5).

During the training process, we explored the option of freezing some or all backbone layers. However, the best results were obtained when training the entire network with a low learning rate of 0.0001, a technique commonly known as fine-tuning (see Table 3 in Section 3).

To optimize the model, we employed the Adam optimizer with a weight decay of 0.01 and utilized the Sparse Categorical Cross Entropy loss function. The model was trained for 40 epochs, and the version with the highest validation accuracy was selected.

Upon successful training, we converted the model to a Tensor RT graph to enhance its performance on the Nvidia Jetson Orin Developer Kit.

2.8 Evaluation with Gradient-weighted Class Activation Mapping

To further assess the model's predictions, we incorporated the Gradient-weighted Class Activation Mapping (GradCam) technique. GradCam identifies and highlights the most important pixels for class-specific classification [Se17]. By utilizing GradCam, we can also detect biases in the dataset. For instance, if the model identifies batteries based on a light reflection rather than the battery itself, GradCam would highlight that reflection.

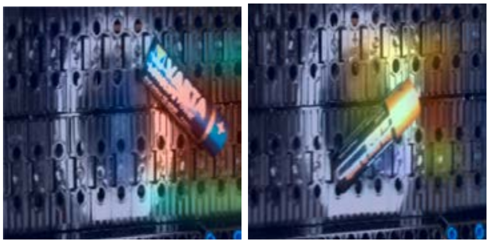


Fig. 7: GRad-Cams, already focusing on battery specific features, blue positive with silver ring (left), copper-coloured positive (right)

The visualized GradCams in Figure 7 demonstrate promising results, with individual features the batteries being prominently highlighted. However, there is still room for improvement as the surrounding area should not be highlighted along with the batteries.

3 Results

The top-performing model achieves 99.21% accuracy on the training dataset and 99.85% on the validation dataset (see Tabular 2). Furthermore, it attains an accuracy of 99.14% (see Tabular 1) on the separate test dataset. In comparison, existing recycling technologies currently achieve a sorting accuracy of 98%. While most of the batteries could be detected with an accuracy of 100% in both datasets, in the validations dataset only Duracell and Ikea batteries are confused and in the separately generated test dataset Duracell, Ikea, Engergizer and High Quality batteries (see Tabular 1 and 2).

In terms of inference time, running the model on Nvidia Jetson AGX Orin takes an average inference time of 0.001-0.002 seconds per image. This enables the classification of a minimum of 50 images per second. Notably, this speed surpasses traditional methods such as X-Ray sensors, which operate at half the rate, and electromagnetic sensors, which are eight times slower. These results were obtained using a dataset consisting of only 3694 images.

Tab. 1: Classification Results for the Test Dataset

Battery	**Precision**	**Recall**	**F1-Score**	**#Images**
Duracell	0.9722	0.9722	0.9722	36
Energizer	0.9487	1.0000	0.9737	37
High Quality	1.0000	0.9630	0.9811	27
Ikea	1.0000	0.9714	0.9855	35
No Battery	1.0000	1.0000	1.0000	27
Topcraft	1.0000	1.0000	1.0000	35
Vatra Alkaline	1.0000	1.0000	1.0000	57
Vatra High Energy	1.0000	1.0000	1.0000	27
Vatra Industrial	1.0000	1.0000	1.0000	38
Vatra Longlife	1.0000	1.0000	1.0000	31
Accuracy			0.9914	350
AVG	0.9921	0.9907	0.9913	350
Weighted AVG	0.9917	0.9914	0.9915	350

Tab. 2: Classification Results for the Validation Dataset

Battery Type	**Precision**	**Recall**	**F1-Score**	**#Images**
Duracell	0.9878	1.0000	0.9939	81
Energizer	1.0000	1.0000	1.0000	48
High Quality	1.0000	1.0000	1.0000	63
Ikea	1.0000	0.9868	0.9934	76
No Battery	1.0000	1.0000	1.0000	50
Topcraft	1.0000	1.0000	1.0000	67
Vatra Alkaline	1.0000	1.0000	1.0000	63
Vatra High Energy	1.0000	1.0000	1.0000	67
Vatra Industrial	1.0000	1.0000	1.0000	64
Vatra Longlife	1.0000	1.0000	1.0000	77
accuracy			0.9985	656
avg	0.9988	0.9987	0.9987	656
weighted avg	0.9985	0.9985	0.9985	656

However, it is important to note that the camera's limitations affect the overall speed. The current webcam utilized in the system can only capture 30 images per second, resulting in slightly slower performance in terms of image acquisition. As such, the camera's speed becomes the limiting factor in achieving greater classification throughput.

Experiments on freezing subsets of the model's backbone during training (see Table 3) show, freezing the entire backbone results in a test dataset performance close to random classification. With 98.28% and 99.39% accuracy on the test and validation datasets, freezing the first half of the model's backbone performs nearly as well as training all layers. However, training the entire model's backbone leads to a 2.28% higher accuracy on the test dataset.

Tab. 3: Classification accuracy when freezing layers of the models backbone during training

#Frozen Layers	Train Accuracy	Val Accuracy	Test Accuracy
154 (all)	0.4678	0.2348	0.1342
77	0.9828	0.9939	0.9686
0	0.9921	0.9985	0.9914

4 Future work

To fully harness the speed potential of the deep learning algorithm, our plan is to utilize a high-speed industrial camera.

In the next phase of our project, we aim to expand our dataset from 10 to 50 different battery series to examine the impact of dataset size on model accuracy. Additionally, we intend to enhance the results of GradCams by not only increasing the number of different battery series but also increase the number of images per battery series.

Due to the chosen image classification approach, the model cannot handle images with multiple batteries. If the batteries are not separated beforehand, the model must be able to handle multiple batteries simultaneously. Therefore, object detection models should be tested for battery detection.

Furthermore, testing under various lighting conditions is necessary to assess whether the model's performance is affected by ambient light.

We have identified 190 distinct nickel metal hydride (NiMH) batteries available on Amazon, representing one out of six chemical compositions for batteries introduced in 2021 [Um23]. Assuming that each of the six battery compositions consists of 190 different batteries, our final dataset would encompass 1,140 unique batteries. Taking into account that NiMH batteries only account for 3% of the market share [Um23] and assuming that the number of different batteries per chemical composition is proportional to the market volume, our final dataset would contain over 6,300 different batteries. Given the increased number of batteries classes, there will be instances where visual features alone are insufficient to distinguish them. Based on our web search we estimate this is only the case for the fewest batteries and think their impact on the final result is negligible.

However if future results on bigger datasets show, that sorting batteries with standard image classification isn't feasible. We will utilize a line scan camera to inspect 100% of the batteries surface for optical character recognition (OCR).
This approach allows us to read the IEC codes of the batteries. Therefore it has potentially a

higher accuracy, but it's also slower and needs a conveyor belt that transports and rotates the batteries.

A combination of text recognition and image classification, where a classifier is trained with the features of a CNN and the text recognition, also seems reasonable, since the manufacturer information is printed on the batteries is an important feature for the classification.

As we expand the dataset not only in terms of the number of classes but also the number of images per class, we anticipate that labeling all the images will become a time-consuming task. Considering the ease of generating images in our use case, we aim to explore the potential of convolutional autoencoders to pre-train a CNN, for image classification, on unlabeled images.

Bibliography

[FM15] Förtsch, Gabi; Meinholz, Heinz: Batterien. In: Handbuch Betriebliche Kreislaufwirtschaft. Springer Fachmedien Wiesbaden, Wiesbaden, pp. 477–500, 2015.

[GBC16] Goodfellow, Ian; Bengio, Yoshua; Courville, Aaron: Deep Learning. MIT Press, 2016. `http://www.deeplearningbook.org`.

[Gi14] Girshick, Ross; Donahue, Jeff; Darrell, Trevor; Malik, Jitendra: Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). June 2014.

[HBF19] Hussain, Mahbub; Bird, Jordan J.; Faria, Diego R.: A Study on CNN Transfer Learning for Image Classification. In (Lotfi, Ahmad; Bouchachia, Hamid; Gegov, Alexander; Langensiepen, Caroline; McGinnity, Martin, eds): Advances in Computational Intelligence Systems. Springer International Publishing, Cham, pp. 191–202, 2019.

[Hi12] Hinton, Geoffrey E.; Srivastava, Nitish; Krizhevsky, Alex; Sutskever, Ilya; Salakhutdinov, Ruslan: Improving neural networks by preventing co-adaptation of feature detectors. CoRR, abs/1207.0580, 2012.

[Ka18] Karbasi, Hamidreza; Sanderson, Adam; Sharifi, Alireza; Pop, Cristian: Robotic Sorting of Used Button Cell Batteries: Utilizing Deep Learning. In: 2018 IEEE Conference on Technologies for Sustainability (SusTech). pp. 1–6, 2018.

[Ki17] Kim, Seunghyeon; Kim, Wooyoung; Noh, Yung-Kyun; Park, Frank C.: Transfer learning for automated optical inspection. In: 2017 International Joint Conference on Neural Networks (IJCNN). pp. 2517–2524, 2017.

[KSH17] Krizhevsky, Alex; Sutskever, Ilya; Hinton, Geoffrey E.: ImageNet Classification with Deep Convolutional Neural Networks. Commun. ACM, 60(6):84–90, may 2017.

[Sa18] Sandler, Mark; Howard, Andrew; Zhu, Menglong; Zhmoginov, Andrey; Chen, Liang-Chieh: MobileNetV2: Inverted Residuals and Linear Bottlenecks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). June 2018.

[Se17] Selvaraju, Ramprasaath R.; Cogswell, Michael; Das, Abhishek; Vedantam, Ramakrishna; Parikh, Devi; Batra, Dhruv: Grad-CAM: Visual Explanations From Deep Networks via Gradient-Based Localization. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV). Oct 2017.

[SGM20] Strubell, Emma; Ganesh, Ananya; McCallum, Andrew: Energy and Policy Considerations for Modern Deep Learning Research. Proceedings of the AAAI Conference on Artificial Intelligence, 34(09):13693–13696, Apr. 2020.

[SK19] Shorten, Connor; Khoshgoftaar, Taghi M: A survey on image data augmentation for deep learning. Journal of big data, 6(1):1–48, 2019.

[Sr14] Srivastava, Nitish; Hinton, Geoffrey; Krizhevsky, Alex; Sutskever, Ilya; Salakhutdinov, Ruslan: Dropout: a simple way to prevent neural networks from overfitting. The journal of machine learning research, 15(1):1929–1958, 2014.

[St21] Sterkens, Wouter; Diaz-Romero, Dillam; Goedemé, Toon; Dewulf, Wim; Peeters, Jef R.: Detection and recognition of batteries on X-Ray images of waste electrical and electronic equipment using deep learning. Resources, Conservation and Recycling, 168:105246, 2021.

[SVL23] Stanford Vision Lab, Stanford University, Princeton University: ImageNet About ImageNet. https://www.image-net.org/about.php, accessed: 22/05/2023.

[Um23] Umweltbundesamt: Altbatterien. https://www.umweltbundesamt.de/daten/ressourcen-abfall/verwertung-entsorgung-ausgewaehlter-abfallarten/altbatterien, accessed: 22/05/2023.