Fine-Grained Product Classification on Leaflet Advertisements

Daniel Ladwig[1], Bianca Lamm[1,2], and Janis Keuper[1]

[1] IMLA, Offenburg University
dladwig@stud.hs-offenburg.de, keuper@imla.ai
[2] Markant Services International GmbH
bianca.lamm@de.markant.com

Abstract. In this paper, we describe a first publicly available fine-grained product recognition dataset based on leaflet images. Using advertisement leaflets, collected over several years from different European retailers, we provide a total of 41.6k manually annotated product images in 832 classes. Further, we investigate three different approaches for this fine-grained product classification task, *Classification by Image*, *Classification by Text*, as well as *Classification by Image and Text*. The last both approaches use the text extracted directly from the leaflet product images. We show, that the combination of image and text as input improves the classification of visual difficult to distinguish products. The final model leads to an accuracy of 96.4% with a Top-3 score of 99.2%.
https://github.com/ladwigd/Leaflet-Product-Classification

Keywords: retail, fine-grained, leaflets, products, image classification, text extraction

1 Introduction

The monitoring of product prices is an important data analysis task for retailers as their own price strategy heavily depends on the prices set by competitors. In this context, the monitoring of product advertisements in printed or online leaflets are the predominant source to obtain pricing and promotion activities from competitors. However, the highly unstructured and multi-modal (image + text information) nature of leaflets and the large number of often very similar products makes the underling product identification and matching task quite challenging. Figure 1 depicts an example of the promotions of the same product in the leaflets of two different retailers.
From a Computer Vision perspective, the retail product price monitoring task resolves into several objectives, from product detection to fine-grained classification (FC). Due to its overall complexity, these tasks are currently mostly solved manually, requiring vast resources. In this work, we focus on the FC task and provide a first dataset with 41.6k manually annotated product images in 832 classes manually obtained from leaflets, alongside first baseline solutions.

2 Related Work

We focus our literature review on publicly available and annotated image collections for the application of *fine-grained product classification* on images in a retail context. The data sources of the datasets vary strongly. First, images from web stores are used

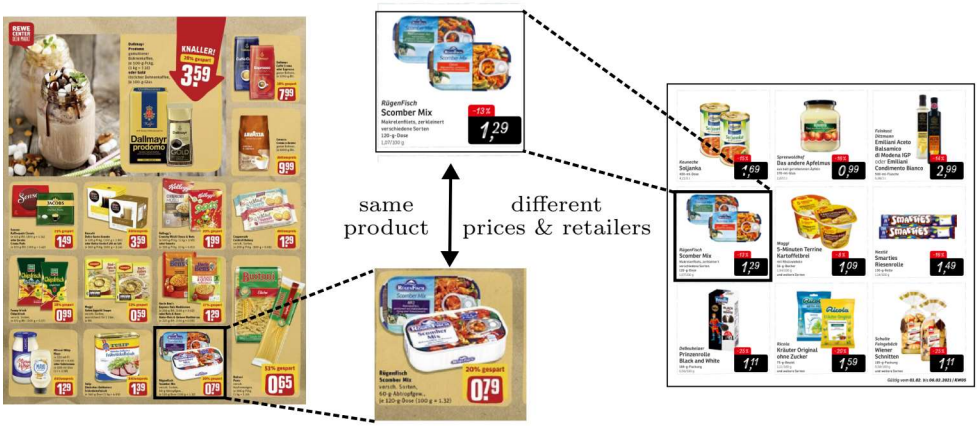

Fig. 1. Price monitoring based on printed leaflets is a key data analysis task in retail, which technically can be defined as a fine-grained, multi-modal classification problem. We provide a first public dataset with 41.6k annotated samples for this task.

as a data source. [1] introduced a *Retail Product Categorisation Dataset* which covers about 48k products with staged "studio" product images in 21 categories. These images are recorded in controlled environments. The authors developed a concatenation of a Convolutional Neural Network (CNN) and a Long Short-Term Network for the FC task [1]. For more realistic real-life scenarios product images "in the wild" are used. The *Products-10k* [2] collection comprises about 10k product classes for about 150k "studio" images and "in the wild" images recorded by customers. The backbone of their approach is the model EfficientNet-B3 [3]. Also, images of supermarket shelves serve as a data source. [4] provides the *Retail-121* dataset consisting of 121 fine-grained retail product categories. For solving the FC task, the authors supply an own approach that is based on a non-parametric probabilistic model and a CNN [4]. To the best of our knowledge, there are no datasets that are based on images from leaflets. Hence, we provide the first annotated dataset containing of product promotions cropped from leaflets.

3 Dataset Description

Data Sources. Our dataset is based on a large collection of full page images in JPG format provided by the company Markant Services International GmbH. The publicly available digital or manually scanned leaflets are circulated by well-known European retailers. Figure 2 shows three representative samples. The leaflets were randomly collected from 132 different[3] retail chains between calendar week 39 in 2016 to calendar week 38 in 2022. The leaflets advertise mainly food and beverages. But also non-food products like household goods, cosmetics, pet foods, or (small) electric devices are promoted. Each leaflet page has been manually segmented into product information boxes by humans. Each box must contain the product image, price, and description. Additional logos, price tags, or quality seals can also be contained. The cropped boxes from the leaflet pages form our provided image dataset. The original leaflets as well as the text information like prices, discounts, or product descriptions are not included in dataset.

[3] Large retail chains often have different subsidiaries and brands which are potentially using the same product images for advertisement. In order to avoid poisoning of the test sets, we grouped all sub-brands into single retailers.

Fig. 2. Three leaflet pages from different European retailers.

Dataset Properties. Each class in the dataset represents one product. The dataset is composed of 832 classes and 41.6k images in total, split into a training set of 33,280 images and a test set of 8,320 images. Each class has 40 images in the training set and 10 images in the test set. All images adhere to a minimal resolution of at least 92 pixels in width and 138 pixels in height while the longer edge is always fixed to 512. This dataset has a size of 3.4 GB. Reducing the longer edge to a fixed value of 256, results in a dataset of a size of 1.2 GB. We provide both versions for public download. Figure 3 shows the training and the test set for a class. The images of a class are similar and diverse at the same time. There can be multiple images from one retailer in a split set but a retailer can only be represented in the training or in the test set for a class.

Fig. 3. Training set (left) and test set (right) of a class.

4 Baseline Solutions

We investigate baseline solutions of the FC task on our dataset by using different information of an image. The *Classification by Image* uses the whole image as input information. The extracted text of an image is the input for the method *Classification by Text*. Moreover, the combination of both information, *Classification by Image and Text*, is analyzed.

4.1 Classification by Image.

We provide first baseline results on our datasets, applying current state-of-the-art methods for FC. We used the dataset version with a longer edge length of 256.
[5] provides a review of current FC methods. For our baseline evaluation, we chose four different image classification models: *ResNet50* [6], *MobileNet V2* [7], a *Vision Transformer* model [8] with a *vit_b_16* architecture and *ConvNeXt* [9] using the base model architecture. Each model has been pretrained on *ImageNet* [10]. The following settings are used for the training process on a NVIDIA GeForce RTX 3090 GPU: 30 epochs of fine-tuning, a batch size of 32, and a SGD optimizer with a learning rate of 0.001 plus a momentum of 0.9. As Table 1 shows, *MobileNet V2* gives the lowest test set accuracy result with an accuracy of 0.894. However, this model needed the shortest training time. The *ResNet50* model and the *Vision Transformer* model have nearly the same test set accuracy of 0.907 and 0.909, respectively. Also the Top-5 accuracy scores of both models is about 0.971. However, the *Vision Transformer* requires almost twice of the training time compared to the *ResNet50* model. The best test set accuracy is reported for the *ConvNeXt* model with 0.921, but its training took almost four hours. We use the *ResNet50* model as the final image model because of its solid accuracy relative to an acceptable training time. In the fine-tuned model the last *Fully Connected* layer is replaced with a *Linear* layer of 2048-1024, followed by a *ReLu* activation function and closing with a *Linear* layer of 1024-832. The hyperparameter used are: batch size of 16 and optimizer momentum of 0.95. A torch ColorJitter is applied as data augmentation with a saturation of 0.5.
Error Analysis. By analysing falsely classified products, errors between visually similar classes occur. Figure 4 exemplifies such products, which still match at a higher abstraction being very similar items from the same producer.

Table 1. Listing of the test accuracy scores, training time, and the Top-5 accuracy for tour FC dataset. The models *ConvNeXt, ResNet50, Vision Transformer* and *MobileNet V2* were trained.

model	test accuracy	training time [h]	Top-5 accuracy
ConvNeXt	0.921	3.9	0.974
Vision Transformer	0.909	2.1	0.971
ResNet50	0.907	1.3	0.971
MobileNet V2	0.894	0.9	0.964

4.2 Classification by Text.

The error analysis of the image model shows the difficulty to distinguish especially between products that are served in different package sizes or product variations that are not clearly visually represented on the product image itself. The leaflet product images mostly include a product description and the serving size. Those texts are not available in extracted digital form yet. To process them in further steps, they need to be extracted from the images first.
OCR Extraction. The Tesseract OCR Engine [11] is chosen to extract texts from the images. This tool combines the text detection and the text recognition in one model.

Different page segmentation modes (PSMs) [4] can be set which alters the way text in the image is treated. Changing modes leads to different extracted text results. Since there is no digital truth text available to evaluate the quality of the extracted text, the accuracy of the classification problem is used to evaluate the quality of the extraction.
To extract much text out of different designed product images with changing background colors, text positions and text sizes, several image preprocessing steps and PSMs are combined. The dataset with the larger image sizes with a longer edge of 512 is used to extract the text to utilize the higher text resolution.
The extraction methods were developed incrementally. An extraction method was put together, the resulting wrong predicted images were inspected and a new fitting method for the poorly performing text extractions was developed. In the final model the following methods were used to extract text from the varying input images. OpenCV[12] and its methods were utilized to import the images and to apply preprocessing. The first four methods use Tesseract with PSMs of 3, 6, 11, 12. Another method applies a grayscaling with cvtColor and cv2.COLOR_BGR2GRAY before applying Tesseract with the standard PSM of 3. For additional methods the gray color change was applied together with a resize of the image times four with Tesseract and PSMs 6 and 11. Finally a grayscaling, resize and cv2.threshold with cv2.THRESH_OTSU, a thresh of 0, maxval 255 and Tesseract with PSM 11 is used. The following preprocessing steps and model are used to evaluate the described methods. They resulted in between 73.6% and 86.4% accuracy.
The combination of all developed extraction methods, to extract as much text from the image as possible, boosts the performance to 91.5%. Even if that means that there are duplicate texts in the collection. Extracting the 33,280 train images sequentially took 16.83h.
Text preprocessing. The extracted texts include wrong letters and signs from noise detected as text in the image. The preprocessing has to be done carefully since package size descriptions can include OCR caused confusions of numbers and letters. In the final solution no stopword or special sign removal is done. The raw extracted texts which include duplicate text from combining the methods are encoded with scikit-learn TfidfVectorizer [13].
Text model. A linear SGDClassifier is trained on the tfidf encoded text and the loss function "modified_huber" with an adaptive learning rate is used. Table 2 shows the model accuracy is 91.5% with a Top-5 score of 96.7%. The text models performance is slightly worse than the image models but the correct predicted images differ. The following step combines those two models.

[4] For further explanation: https://github.com/tesseract-ocr/tessdoc

Fig. 4. Illustration of two similar products. The left product is often confused with the right one by three of the four baselines.

4.3 Classification by Image and Text.

The first step to combine the text and image models is to apply a softmax to each probability predictions. After that a weighted probability stacking is done by giving the text model a higher weight, as the SGDClassifier predicts way lower decision probabilities than the ResNet50 model. Weighting the text model allows to make an impact after stacking the probabilities. The final predictions are differentiated in predictions where both models agree (labeled as high confidence prediction) and predictions where they do not (labeled as low confidence prediction). This allows to separate those low confidence predictions to be potentially reviewed manually with the help of a Top-3 selection with a very high accuracy of 99.2%. Figure 5 displays the process of combining the models and the resulting predictions.

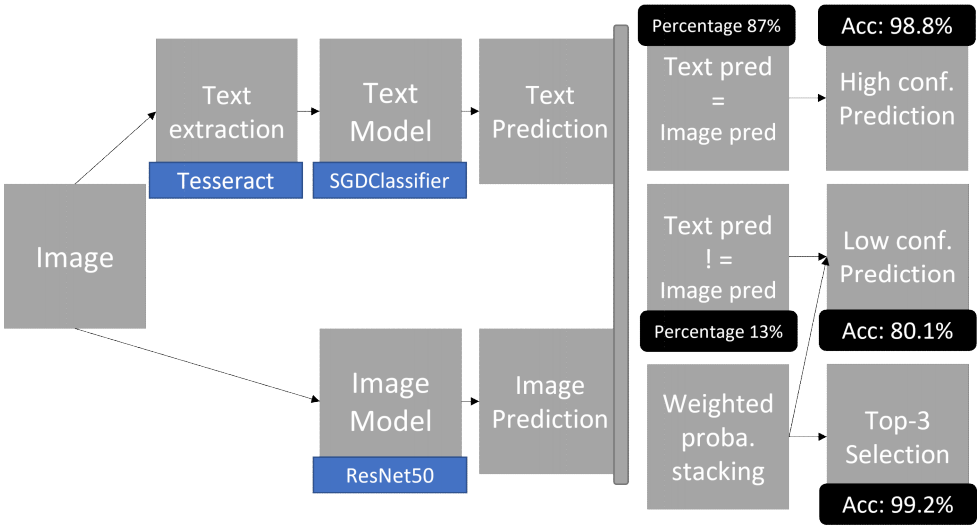

Fig. 5. Model Combination Process

Combination Results. The prediction stacking method by combining the probabilities of the text and image model results in an overall accuracy of 96.4% which is a significant increase from the models alone. Interesting is the calculated Top-3 accuracy of 99.2%. This shows the ability of the combined models to find the correct class in the fine-grained problem and rate it high. The gap between the prediction and Top-3 selection displays the difficulty to make the correct choice when the products do not differ much, sometimes only one attribute differentiates them. There is still space for improvement in the combination process of the two models. The accuracy that the image model prediction or the text model prediction is correct lays by 98.0% which is not quite reached by the used weighted probability stacking method.

Error Analysis. A manual review of the wrong predicted images and classes of the final combined model revealed the following findings. As in the beginning presumed the products that are served in different variations or package sizes can cause confusions for the final prediction. The high Top-5 score shows that the correct product can be found but the final choice which variation is presented is still a difficult task. Figure 4 already shows an example of two similar product servings. Figure 6 shows two classes of products that are distinguished by serving size. The Top-3 products are served in 250g and the bottom products are served in 290g. The package size displayed on the product varies, even within the class. This explains the confusion between those classes for the image

Table 2. Model and Combination Results

Model	Method	Accuracy	Top-3	Top-5
Final ResNet50	Image	0.925	0.961	0.969
SGDClassifier	Text	0.915	0.962	0.967
Combined	Weighted proba.	0.964	0.992	0.993

model. The text model could possibly distinguish those two from the given package size. Although the last image labels the package size as "250g + 40g" instead of the expected 290g which can lead to confusions. In this case the combined model could predict both classes correctly, but this is an example how difficult the differentiation between similar classes can be. Changing product descriptions or difficult text extractions impact the text model predictions negatively.

Fig. 6. Model Confusion Example

5 Conclusions

In this paper, we provide a dataset and present a baseline solution for fine-grained image classification of retail products on leaflets. We propose methods to optimize the product classification by combining image classification with text classification based on text directly extracted from the images. The presented methods allow to extract and process text from varying product images. Improved results compared to the image classification alone support the idea to make use of existing text. Combining text and image classification improves the ability to distinguish between visually similar product representations. As a first part of the price monitoring task we create a sufficient baseline for product matching. Future work will be the extracting of the prices of the product promotions. This is challenging due to the huge variance in color, size, and position in the promotion.

References

1. Elayanithottathil, F.S., Keuper, J.: A retail product categorisation dataset. arXiv preprint arXiv:2103.13864 (2021)
2. Bai, Y., Chen, Y., Yu, W., Wang, L., Zhang, W.: Products-10k: A large-scale product recognition dataset. arXiv preprint arXiv:2008.10545 (2020)
3. Tan, M., Le, Q.: Efficientnet: Rethinking model scaling for convolutional neural networks. In: International conference on machine learning, PMLR (2019) 6105–6114
4. Karlinsky, L., Shtok, J., Tzur, Y., Tzadok, A.: Fine-grained recognition of thousands of object categories with single-example training. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2017) 4113–4122
5. Wei, X.S., Song, Y.Z., Mac Aodha, O., Wu, J., Peng, Y., Tang, J., Yang, J., Belongie, S.: Fine-grained image analysis with deep learning: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence (2021)
6. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2016) 770–778
7. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C.: Mobilenetv2: Inverted residuals and linear bottlenecks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2018) 4510–4520
8. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)
9. Liu, Z., Mao, H., Wu, C.Y., Feichtenhofer, C., Darrell, T., Xie, S.: A convnet for the 2020s. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2022) 11976–11986
10. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition, Ieee (2009) 248–255
11. Smith, R.: An overview of the tesseract ocr engine. In: Ninth international conference on document analysis and recognition (ICDAR 2007). Volume 2., IEEE (2007) 629–633
12. Bradski, G.: The OpenCV Library. Dr. Dobb's Journal of Software Tools (2000)
13. : sklearn.feature_extraction.text.tfidfvectorizer. https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html