

# A Visual RAG Pipeline for Few-Shot Fine-Grained Product Classification

Bianca Lamm  
Markant Services International GmbH  
Bianca.Lamm@de.markant.com

Janis Keuper  
IMLA, Offenburg University and  
University of Mannheim  
keuper@imla.ai

## Abstract

*Despite the rapid evolution of learning and computer vision algorithms, Fine-Grained Classification (FGC) still poses an open problem in many practically relevant applications. In the retail domain, for example, the identification of fast changing and visually highly similar products and their properties are key to automated price-monitoring and product recommendation.*

*This paper presents a novel Visual RAG pipeline that combines the Retrieval Augmented Generation (RAG) approach and Vision Language Models (VLMs) for few-shot FGC. This Visual RAG pipeline extracts product and promotion data in advertisement leaflets from various retailers and simultaneously predicts fine-grained product ids along with price and discount information. Compared to previous approaches, the key characteristic of the Visual RAG pipeline is that it allows the prediction of novel products without re-training, simply by adding a few class samples to the RAG database.*

*Comparing several VLM back-ends like GPT-4o [23], GPT-4o-mini [24], and Gemini 2.0 Flash [10], our approach achieves 86.8% accuracy on a diverse dataset.*

## 1. Introduction

The task of Fine-Grained Classification (FGC) enables a detailed analysis and more precise categorization of items that are highly similar in nature. The differentiation of items arises in several areas, such as animals, plants, cars, and retail products [29]. In the retail domain, traded products are continuously changing, while thousands of products are sold in a single supermarket. Identical products are identified by the standardized and unique Global Trade Item Number (GTIN) [2]. Hence, the prediction of GTINs is defined as a fine-grained classification task. Products differ in a fine-grained manner, *e.g.*, due to similar content but different packaging size. The previous methods used for FGC are mostly based on Convolutional Neural Networks or Vision Transformers [7], which provide a

reasonable accuracy but have to be retrained very often in practice in order to keep up with the fast changing product portfolios. In recent years, Vision Language Models (VLMs) have significantly advanced the integration of visual and textual data, enabling more sophisticated multi-modal understanding [33]. These models have the capability for solving complex reasoning tasks, including image captioning and visual question answering [28, 33]. So far, VLMs have mainly a knowledge about public available data [33], which typically does not include detailed information about retail products. The Retrieval Augmented Generation (RAG) approach shows promising outcomes by incorporating external knowledge sources [9]. Recently, the emergence of multi-modal inputs for RAG is an ongoing research topic [1]. The creation of a custom database which is required for the RAG method offers the ability to provide the necessary context to VLMs for specific tasks.

In this paper, we investigate the combination of multi-modal RAG with VLMs for a few-shot fine-grained classification of retail products and their properties from advertisement leaflets. Our approach allows to add novel products to the FGC pipeline, simply by adding a few samples to the RAG database. This re-training free approach achieves state-of-the-art accuracies on a multi-modal benchmark. Our introduced Visual RAG pipeline is a novel approach to RAG using VLMs. The creation of a context, which is part of the VLM prompt, is essential for facilitating the VLMs comprehension of the task. In addition, the investigation of the RAG approach on visual data from the retail domain constitutes a novel methodology toward FGC.

## 2. Related Work

An overview of the topics FGC, image datasets in retail, and RAG is presented in the Sections 2.1 to 2.3.

### 2.1. Fine-Grained Classification

The task of FGC is presented in different domains in particular on image classification. An example is the image

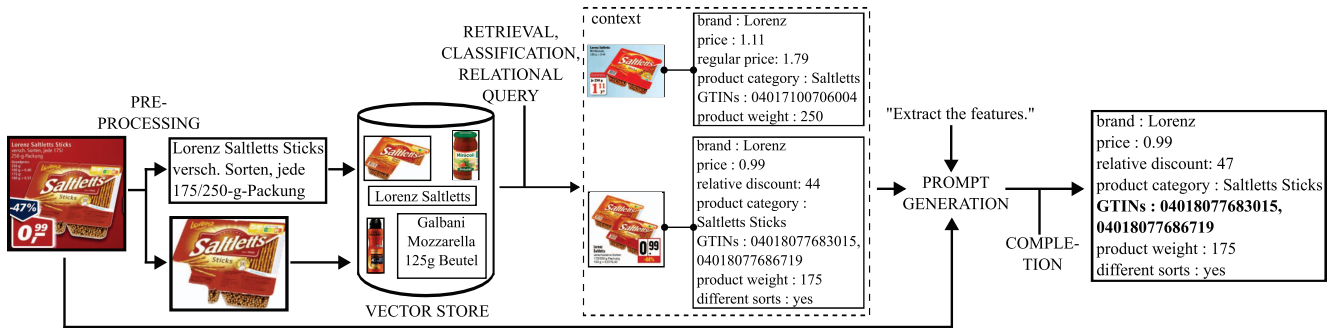


Figure 1. Illustration of the presented Visual RAG pipeline. The pipeline is based on the RAG approach and is characterized by five main steps: *Preprocessing*; *Vector Store*; *Retrieval, Classification, Relational Query*; *Prompt Generation*; and *Completion*. Moreover, a contextual knowledge comprising few-shot samples with corresponding task solutions is appended to the prompt for the employed VLM. The prediction of the target *GTINs* serves as FGC. The additional predictions deliver to enrich the objectives.

classification of bird species [29]. [21] provides a dataset of 120k images split in 60 classes. Further information about bounding boxes, segmentation masks, or habitat environment are provided. The authors of [17] investigate the classification of car models. This image dataset contains about 16k images divided in 197 classes. The image dataset LZUPSD [32] consists of about 4.5k images. The images show plant seeds from 88 different seeds.

## 2.2. Image Datasets in Retail

There are thousands of retail products among which individuals are difficult to distinguish due to, *e.g.*, the packaging. The work of [4] presents a dataset consisting of about 1k images. The images were taken under studio conditions and show single food products from different views. In addition, class labels and object detection labels are provided. The Grocery Store Image Dataset [16] combines retail products and the task of FGC. The objects on the images of the dataset are fruits and vegetables as well as dairy and juice products. The dataset contains about 5.1k images divided in 81 fine-grained classes. Per class an iconic image and multiple natural images are collected. The authors of [34] provide an image dataset that comprises the pages of IKEA catalogs. On basis of this data, the IKEA dataset [3] is created. The dataset contains about 9.5k images and almost 24k texts extracted from the aforementioned pages. The Retail-786k [18] image dataset consists of about 786k images split into about 3k classes. The images show product advertisements cropped from leaflet pages. This dataset forms the basis of the investigations of this paper.

## 2.3. Retrieval Augmented Generation

The RAG method derives from the Natural Language Processing (NLP) [20]. This approach combines data retrieval and text generation by accessing and using external knowledge [20]. Recently, multi-modal RAG approaches extend the classic RAG method by using other modalities, such as

text, images, audio, or video. A comprehensive overview of datasets and benchmarks that evaluate multi-modal RAG approach is shown in [1]. [30] introduces the retrieval and generation of text and images. [31] presents a VLM-based RAG pipeline, called VisRAG. Furthermore, the authors of [27] expand the query input to the VLM with the retrieved text and image samples. The improvement of the retrieved samples is evaluated in [5] by suggesting a knowledge-enhanced reranking and noise-injected training.

## 3. Dataset

We use a subset of the *Retail-786k* [18] image dataset, supplemented with additional textual data per image. The addition of textual data to the image dataset enables more complex investigations to be carried out. Section 3.1 provides a detailed description of the images. In Section 3.2, the thorough explanation of the text information are presented. The unique characterization of dataset used is the incorporation of textual data pertaining to products and promotions. Figure 2 illustrates an item of the dataset that consists of an image and its product and promotion data. The dataset is published on the website: [https://huggingface.co/datasets/blamm/retail\\_visual\\_rag\\_pipeline](https://huggingface.co/datasets/blamm/retail_visual_rag_pipeline).

### 3.1. Image Data

The dataset used consists of 4,771 images labeled into 367 different classes. The images are split into subsets of 3,670 training images and 1,101 test images. The dataset is balanced but the high number of classes with relatively few examples per class makes the FGC task very challenging. Each training class has 10 images, each test class has 3 images. The images have a size of 512 pixels on the longer edge (see Figure 2a for an example).

### 3.2. Product and Promotion Data

For additional details, the data about the product and promotion is made accessible. The product information comprises a detailed account of product properties: *brand*, *product category*, *GTINs*, *product weight*, and *different sorts*. In the case that a promotion covers a variety of different types or flavors of the product, the GTIN of each type is recorded. It is common practice for promotional pricing to include not only the special-offer price of the product, but also the regular price and/or the discount. Discounts can be classified as either relative or absolute. Hence, the characterization of a promotion includes the prediction targets: *price*, *regular price*, and *relative discount* or *absolute discount* additional to the FGC of the product GTINs (see Figure 2b for an example of product and promotion data). The importance of the target *GTINs* is given by the unique identifier for products. A promotion image can have assigned either a single GTIN or multiple GTINs. A multitude of GTINs originates by the promotion of products with different flavors or different weight quantity. In the retail and supplier domain, the data about GTINs are essential for reporting and analysis.

### 3.3. Fine-Grained Dataset

The fine-grained characterization of the dataset is presented in a variety of forms. Firstly, the assessment may depend on the image. Advertisements of products with different weights illustrate a high visual similarity. Figure 3 shows such a similarity although the images belong to different classes due to the product weights. Moreover, product advertisements for the same product but from different brands indicate visual similarity, too. Figure 4 illustrates this fine-grained emergence. In addition, there are product advertisements that show different products of the same brand. These promotions also demonstrate a fine-grained distinction. Figure 5 illustrates advertisements of the different products, cereals and cereal bars. The products derive from the same brand. Hence, the product packaging exhibit a strong visual similarity.

Even the product and promotion data manifest resemblance in the dataset. Table 1 presents two items of the dataset that are only distinguished by the value of *GTINs*, which is not visible in the corresponding image, and by the divergent value of *product weight*.

## 4. Visual RAG Pipeline

Figure 1 illustrates the proposed Visual RAG pipeline, consisting of five main steps: The first step is defined as *Pre-processing* described in Section 4.1. In this step, the query image will be preprocessed to obtain the data that are stored in the *Vector Store*. The structure of the *Vector Store* and its stored data is elucidated in Section 4.2. The subsequent step is called as *Retrieval*, *Classification*, *Relational Query*,



(a) Image from the dataset.

data type	target	value
product	brand	Lorenz
	product category	Saltlets Sticks
	GTINs	04018077683015, 04018077686719
	product weight	175.0 Gramm
	different sorts	yes
promotion	price	0.99
	regular price	NaN
	relative discount	47
	absolute discount	NaN

(b) Product and promotion data. Missing target values are stored as NaN.

Figure 2. Illustration of an item in the dataset that consists of an image (2a) and textual product and promotion data (2b).



(a) Product weight is 400g.

(b) Product weight is 750g.

Figure 3. Illustrations of images that demonstrate a fine-grained difference due to variations in product weight. The evaluations of ResNet50 [11] and BERT [15] show misclassification of these images.

which is outlined in Section 4.3. In this step, a retrieval is performed, followed by a classification being conducted using the retrieved data. Based on the classification the step *Relational Query* is executed. In this step a context of few-shot samples is created. The following step includes the *Prompt Generation* presented in Section 4.4. The last step of the pipeline is called *Completion*, which executes a re-



(a) Product by brand "Öttinger".



(b) Product by brand "Maisele's".

Figure 4. Illustrations of images that demonstrate a fine-grained difference due to the products are from different brands. The evaluation of ResNet50 [11] reveals the misclassification of these images.



(a) The product is cereal.



(b) The product is cereal bars.

Figure 5. Illustrations of images that demonstrate the fine-grained differences between the products. The differences are in the products themselves, even though they are from the same brand.

target	item 1	item 2
brand	Heinz	Heinz
product category	Tomato Ketchup	Tomato Ketchup
GTINs	08715700017006	00000087157215
product weight	500.0 Milliliter	400.0 Milliliter
different sorts	yes	yes
price	1.99	1.99
regular price	2.49	2.49
relative discount	20	20
absolute discount	NaN	NaN

Table 1. Illustration of the product and promotion data for two dataset items differing only in *GTINs* and *product weight*.

quest to a VLM considering the generated prompt. A detailed description is defined in Section 4.5.

#### 4.1. Preprocessing

The input to the pipeline is an image as depicted in Figure 2a. The image is processed in two different ways de-

pending on the data type stored in the *Vector Store*. Initially, the tool Language Segment Anything (LangSAM) [22] is applied to the query image. The result is a segmentation mask that is described by a given prompt. We use the prompt: "product.". For the image data that is stored in the *Vector Store*, the segmentation mask is cropped from the query image. The left image of Figure 6 depicts the product image of the query image and is saved in the *Vector Store*. The product description printed on the advertisement image is also stored in the *Vector Store*, but as text data. To obtain this text, another preprocessing step is necessary. This time, the segmentation mask is eliminated from the query image, showing in Figure 6 through the image "Image without product". The VLM GPT-4o [23] (version: 2024-08-06) is used for the extraction of the textual product description. The system message is defined by: "You are an AI assistant that extract text from an image". The user prompt includes the preprocessed image without the product as well as the task description: "First, extract the text. Second, remove all price information. If available, remove all special / detailed description of the product".

#### 4.2. Vector Store

We use the Chroma vector database [14] as *Vector Store* in our pipeline. We add images and texts into *Vector Store*. The left image and the right text of Figure 6 show exemplary data that are stored. 5,390 product images and 3,670 product description texts are stored in the *Vector Store*. The embedding vectors in the *Vector Store* are created by the *OpenCLIPEmbeddings* [13]. For the text tokenizer, the model *xlm-roberta-large-ViT-H-14* is used. The similarity search in the *Vector Store* is performed via cosine distance. Each embedding vector stores additionally the information of the class label as meta data.

#### 4.3. Retrieval, Classification, Relational Query

For the step *Retrieval*, the five most similar embedding vectors to the query embedding are returned. Each embedding vector has stored the class label. Therefore, the following *Classification* step determines the most frequently occurring class label. In cases of ties, the class label of the nearest image embedding vector is returned. Based on the additional label information, the embedding vectors can be filtered according to the classified label. In the subsequent step *Relational Query*, the corresponding advertisement image as well as the product and promotion data of the filtered embeddings vectors are delivered by an external database. This information, images and texts, serves as few-shot samples and forms the contextual knowledge that is a part of the *Prompt Generation*.

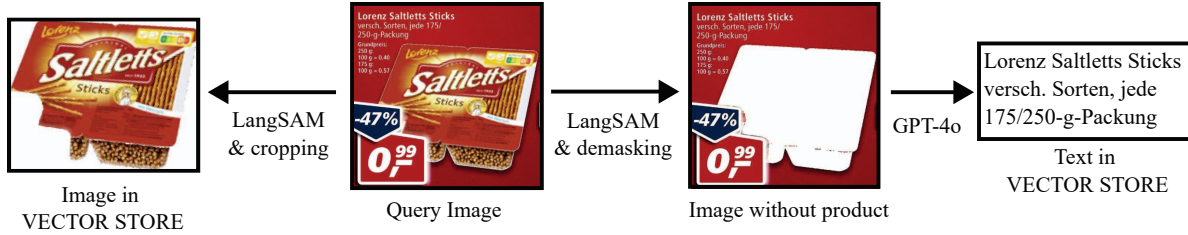


Figure 6. Illustration of the *Preprocessing* step of the Visual RAG pipeline. The results of the *Preprocessing* steps are stored in the *Vector Store*. The images stored in *Vector Store* are produced by the tool LangSAM [22] and image cropping. The result of using the tool LangSAM [22] and demasking is the input for the VLM GPT-4o [23] getting the product description.

#### 4.4. Prompt Generation

The prompt given to the VLM is divided into three parts: task, query image, and context. The task is formulated as: "Extract all features". The features, also known as targets, refer to the product and promotion data. As a further part of the prompt, a contextual knowledge is provided. The context is understood as data consisting multiple few-shot samples, containing product images along with their corresponding product and promotion data. The maximum number of samples is set to three, and a minimum of one sample is to be provided. The number of samples depends on the VLM used. Should the maximum input token length be exceeded by the samples, then the number of samples is reduced accordingly. The number of data examples included in the context must be limited depending on the VLM due to the maximum input token length of the model. The prompt is structured, beginning with the task description, followed by the contextual knowledge containing few-shot samples, and concluding with the query image.

#### 4.5. Completion

The *Completion* step comprises the VLM request. The input for the VLM is the prompt described in the previous section. We investigate the VLMs: *GPT-4o* [23] of the version 2024-08-06 (*GPT-4o\_2024-08-06*), *GPT-4o-mini* [24] of the version 2024-07-18 (*GPT-4o-mini\_2024-07-18*), and *Gemini-2.0-flash* [10]. The models *GPT-4o* and *GPT-4o-mini* have a maximum input token length of 128,000 [23]. These models offers the usage of structured output [25] that is used. The object schema is defined by using Pydantic [6]. Each target is defined as an attribute of the Pydantic model with appropriate data types. Section 8 in the Appendix shows the structured output of the VLM *GPT-4o-mini\_2024-07-18* for the query image shown in Figure 2.

### 5. Evaluation

In Section 5.1, the various baseline evaluation procedures and the presented Visual RAG pipeline are investigated in relation to the FGC task. Further examinations are described in Section 5.2.

#### 5.1. Evaluation of Fine-Grained Classification

For a baseline evaluation of the FGC task, we investigate image only, explained in Section 5.1.1, text only, described in Section 5.1.2, and combined image-text models, so-called multi-modal models, depicted in Section 5.1.3. Moreover, the evaluation of the Visual RAG pipeline in relation to the FGC task is illustrated in Section 5.1.4.

The examinations in Sections 5.1.1 and 5.1.2 were conducted using a single NVIDIA Tesla T4 GPU. The described investigations in Section 5.1.3 and all evaluations of the Visual RAG pipeline are executed on a single CPU.

The evaluation of the FGC task is based on the product target *GTINs*. A class of the dataset consists, among other things, of product advertisement images. An image of a class can, e.g., advertise a single product, whilst other images of the same class promote multiple varieties of the same product. Therefore, several *GTINs* are assigned to these images. A class of the dataset is defined by the union of the target *GTINs* values. This forms the basis of the evaluation of the target *GTINs* in Table 2.

##### 5.1.1. Image Classification

For a baseline image-only classification, the model ResNet50 [11] is used. The configuration settings for the training are: epochs = 50, learning rate = 0.001, batch size = 32, optimizer = SGD, momentum of optimizer = 0.9, and image resize size = 256. The duration of the training is almost two hours. The evaluation of the trained model on the test dataset yields to an accuracy score of **84.4%**. Examples for misclassified images are shown in Figures 3 and 4.

##### 5.1.2. Text Classification

For a baseline text-only classification, the description text on the images is used. We follow the text extraction method described in [19], using the OCR tool PyTesseract [12]. An image with the corresponding OCR-extracted product description text is included in Section 7 in the Appendix. The text classification is based on the model BERT [15]. The tokenizer and the classification model are provided by the

Hugging Face transformers library [8]. We use the model versions: *bert-base-uncased* and *BertForSequenceClassification*. The configuration settings for the tokenizer and the classification model training are: tokenizer maximum length = 128, tokenizer padding = max length, epochs = 30, learning rate =  $2e-5$ , batch size 32, and optimizer = AdamW. The training duration is more than half an hour. The evaluation of the trained model on the test dataset yields to an accuracy score of **74.2%**. Examples for false classified text shows the following: OCR-extracted text of test data like "*SÜDZUCKER Puder Zucker Mühle\* je 250-g-Dose (100 Q = 0.40)*" or "*tsmengen Un SÜDZUCKER Puder Zucker Mühle\* je 250-g-Dose (100 g = 0.40)*" are misclassified into the class containing training data like "*Diamant Gelierzucker 1:1 1-kg-Packung*", "*Diamant Gelierzucker In E C TTT*", and "*Diamantb Gellerzucker 11 1-kg-Packung*". The classification by the OCR-extracted text of images, displayed in Figure 3b, are misclassified into the class, illustrated in Figure 3a.

### 5.1.3. Multi-modal Classification with CLIP

The CLIP [26] model is based on the concept by learning from text-image pairs. In particular, CLIP allows zero-shot classifications. On the basis of these skills, we investigate the CLIP model as zero-shot baseline. Specially, the model version *ViT-B/32* is used. In order to use zero-shot learning, it is necessary to provide a textual description of the classes. In our case, the text is build by joining the prediction targets *brand* and *product category*. The emphasis was placed on these two targets, as the *brand* is typically visible in the image and the *product category* constitutes an additional specification of the product. The *product category* can be utilized to differentiate between different products of the same *brand*. Moreover, the CLIP model [26] has a context length limitation. As example for the image in Figure 2a the class description is: "*Lorenz - Saltletts Sticks*". The accuracy score of the CLIP model is about **45.9%**. This score is below the image and text classification. However, there is neither training process nor costs associated with the usage of zero-shot learning on the CLIP model.

### 5.1.4. Classification using our Visual RAG pipeline

The Visual RAG pipeline returns a prediction per each target. This also applies to the target *GTINs* according to which the FGC task is defined. The classification using the Visual RAG pipeline is determined if the predicted *GTINs* value is included in the *GTINs* union of a class. Hence, a comparison of the baseline experiments can be made. The examination of the Visual RAG pipeline using the VLM *GPT-4o-mini\_2024-07-18* shows an accuracy score of **86.8%**.

Table 2 shows the results of Image Classification, Text Classification, Multi-modal Classification, and Classification using the Visual RAG pipeline according to FGC.

The image-only classification with ResNet50[11] performs with an accuracy score of 84.4%. Text Classification and Multi-modal Classification are unable to approximate to the aforementioned value. The accuracy scores are 74.5% and 45.9%, respectively. The Classification using the Visual RAG pipeline outperforms the other methods by achieving an accuracy score of 86.8%. In addition, the Visual RAG pipeline provides a more comprehensive output which can be useful for further tasks. The fact that there is no requirement for a model to be trained is a significant advantage of the Visual RAG pipeline. There is a constant change of products in retail, so frequent training of models for image and text classification is necessary.

Classification	Model	GTINs
Baseline Image	ResNet50[11]	84.4%
Baseline Text	BERT[15]	74.5%
Baseline Multi-modal	CLIP[26]	45.9%
Visual RAG pipeline	GPT-4o-mini_2024-07-18[24]	<b>86.8%</b>

Table 2. Illustration of the accuracy scores of the FGC task represented by the *GTINs*. The evaluation of target *GTINs* is based on the set defined by the class, which is understood as the union of the GT values of this target.

## 5.2. Auxiliary Analysis

Further analysis pertaining to the Visual RAG pipeline are contemplated. Specifically, the impact of the VLM and the contextual knowledge are discussed in Sections 5.2.1 and 5.2.2, respectively. In Section 5.2.3 an alternative evaluation measure of the target *GTINs* is described. Subsequently, the evaluation of context biases and false predictions as well as the value and cost analysis follow in Sections 5.2.4 and 5.2.5.

The Visual RAG pipeline has been developed to make open-ended predictions across multiple product property targets, like *price*, *discount*, or *brand* as introduced in Section 3.3. The method of validating the accuracy score of a target in the Visual RAG pipeline, depends on the target itself. For the targets *brand* and *product category* it is sufficient if the predicted value is a substring of the Ground Truth (GT) value. Examples are as follows: the predicted *brand* "*LOreal*" is assessed correctly due to the GT value "*[LOreal, Men Expert]*" as well as the predicted *product category* "*Pastasauce*" and the GT value "*[Nudelsauce, Pasta Sauce, Pastasauce, Pasta-Sauce]*". The validity of this validation method is substantiated by the observation that the same product is advertised using different spellings for *brand* and/or *product category*. For the other targets, the prediction has to be equal. The requirement for identical *GTINs* makes it possible to identify the promoted product on the advertisement images in a more specific way.

### 5.2.1. Ablation Study: Impact of the VL-Model

The impact of VLMs in the Visual RAG pipeline is evaluated by incorporating the accuracy of all predicted targets. The VLMs *GPT-4o-mini\_2024-07-18* [24], *GPT-4o\_2024-08-06* [23], and *Gemini-2.0-flash* [10] are investigated. Table 3 shows the accuracy score per target for the examined VLMs. The variance in accuracy of the target *GTINs* is significant. The evaluation of this target is based on an equal numerical value of the prediction and GT. The models *Gemini-2.0-flash* and *GPT-4o\_2024-08-06* reach 75.4% and 76.5% accuracy, respectively. The model *GPT-4o-mini\_2024-07-18* increases the accuracy score to 81.2%.

### 5.2.2. Ablation Study: Impact of contextual knowledge

The false predictions of the target *GTINs* is explained in more detail. In total, 206 of 1,101 test images receive a false predicted *GTINs*. The most of these images receive a NULL value for the target *GTINs*. Due to the fact that the predictions for all other targets are also NULL, it suggests that the VLM response is invalid in these cases. Further investigation has been conducted to deepen the understanding of the invalid response of the VLM. The structure of the context containing few-shot samples is based on data similar to the query image. The quantity of samples is predetermined at this stage. In order to obtain a valid response, the context is reduced to contain only a single sample instead of multiple samples. This procedure is applied using the VLM *GPT-4o-mini\_2024-07-18*. Hence, this approach achieves the best accuracy score of **86.0%** for the target *GTINs*, shown in the last column of Table 3. For this evaluation, the exact match of the GT value and prediction was applied instead of the metric used in Section 5.1.

### 5.2.3. Alternative evaluation measure

So far, our evaluation for the target *GTINs* requires predictions to be equal to the GT values, even though the GT value can comprise a list of *GTINs*. A product advertisement often promotes multiple products using a single representative product image, *e.g.*, different flavors of the same base product. This results in numerous *GTINs* being assigned to the image. A single correctly predicted *GTIN* should be recognized as a reference point. For this reason, another evaluation method was applied. The evaluation method is as follows: the prediction is considered as valid if at least a single predicted *GTIN* is correct. According to this error analysis, the Visual RAG pipeline with the VLM *GPT-4o-mini\_2024-07-18* using the method of reducing contextual knowledge elevates the accuracy score to **92.3%**.

### 5.2.4. Context Biases and False Predictions

Introducing few-shot context into the prediction pipeline also has some drawbacks, as biased context can lead to false predictions. In the following, we summarize context biases we could observe in our evaluation. Please refer to Section 9

in the Appendix for visualizations of the described cases.

**Image-Text Bias.** In some cases, the models ignore the correct contextual information regarding the *GTIN* and extract arbitrary numbers which are visible in the query image instead. Figure 9 in the Appendix shows an example for this behavior. A possible mitigation for this bias is to specifically instruct the VLM not to use printed numbers in the query image to predict the *GTIN*.

**Image-Context Bias.** Figure 7 shows an advertisement image that promotes only a single product. The hint that the price is valid for different sorts is not printed in the image. Hence, the GT value for the target *GTINs* only consists of a single number. However, the context of few-shot samples provided for the VLM contained images promoting different sorts of the product. Therefore, the product data of the context comprise a list of *GTINs* per each context image. Basically, the quality of the GT values for the target *GTINs* is considered critically. There are images in the dataset that promote a single product without the additional note of validity for different sorts. However, the product target *GTINs* contains a list of numbers. Also, the opposite appears: the advertisement image promotes a product in different sorts but only a single *GTIN* is stored in the data.

**Database Bias.** Due to marketing impacts, some advertisements have displayed more than one price in the image. It is also the case that the printed pricing information is not the unit of the promoted product. For example, an advertisement promotes for a carton of six bottles. However, the price detail is per bottle and the GT value in the dataset is defined as price for the carton with six bottles.

**Missing Data Bias.** If the advertisement image does not display the information of regular price, the prediction is taken from the context containing few-shot samples of similar data. Further, the indication of the recommended retail price is false predicted as regular price. The recommended retail price is basically not defined as the GT for the target *regular price*. Other false predictions for the named target include the declaration about the reference weight price. A false prediction of this kind happens especially if there is no specification about regular price in the image.

### 5.2.5. Value and Cost Analysis

The cost analysis depends on the token numbers. The number of FLOPs is not specifiable as only commercial models are used. All steps of the Visual RAG pipeline are executed on a single CPU. The average duration of the *Completion* step of the VLM *GPT-4o-mini\_2024-07-18* is about 7.9 seconds. The average costs per a single *Completion* step is about \$0.01. Hence, the total costs for the test set amounts to \$15.28. Further costs may arise due to the allocation of the VLM and the RAG database. Table 4 provides an overview of various parameters for each VLM investigated.

data type	target	GPT-4o-mini_2024-07-18	GPT-4o_2024-08-06	Gemini-2.0-flash	GPT-4o-mini_2024-07-18 <sup>+</sup>
product	brand	90.8%	86.2%	90.7%	<b>96.8%</b>
	product category	88.1%	82.6%	<b>93.0%</b>	<b>93.0%</b>
	product weight	80.0%	77.7%	<b>85.2%</b>	84.7%
	GTINs	81.2%	76.5%	75.4%	<b>86.0%</b>
	different sorts	<b>49.4%</b>	48.9%	49.0%	<b>49.4%</b>
promotion	price	87.9%	87.8%	91.5%	<b>91.8%</b>
	regular price	36.5%	<b>49.2%</b>	47.6%	32.0%
	relative discount	32.5%	<b>38.6%</b>	31.8%	30.4%
	absolute discount	89.4%	<b>94.3%</b>	81.0%	88.2%

Table 3. Illustration of the accuracy scores per target for each examined VLM. The test dataset comprises of 1,101 images. The evaluation of target *GTINs* is based on an equal numerical value of the prediction and GT. <sup>+</sup>: If the VLM response exclusively contains NULL values then further requests with reduced context is executed.

parameter	GPT-4o-mini_2024-07-18	GPT-4o_2024-08-06	Gemini-2.0-flash
avg. input tokens	92,888	88,015	104,119*
avg. output tokens	90	87	316*
avg. total tokens	92,978	88,102	104,436
avg. elapsed time/req.[s]	7.9	10.2	4.8
approx. total costs	\$15.28	\$241.91	\$10.98

Table 4. Illustration of various parameters for each examined VLM. \*: Data of prompt and candidates tokens, respectively.



	GTINs
GT	07613034229083
prediction	07613034228673, 07613034228826, 07613034229083

Figure 7. Illustration of an image plus GT and prediction values for the target *GTINs*. The prediction comprises a list of *GTINs* due to the fact that the data in the context exclusively promote for different sorts of the product.

## 6. Limitations and Outlook

This paper introduces a dataset consisting of images as well as product and promotion data per image. With the definition of the prediction target *GTINs*, the dataset is investigated for the task of FGC. Furthermore, we present a Visual RAG pipeline containing of five main steps. The results of the *Preprocessing* step serve as input for the *Vector Store*. The composition of the context that consists of few-shot samples (image, product and promotion data) is produced by the step *Retrieval, Classification, Relational Query*. The *Prompt Generation* step joins the query image, the context, and the task description. The result of the pipeline or the output of the *Completion* delivers the product and promo-

tion data for the query image. The efficacy of the entire pipeline is contingent upon the quality of the segmentation mask, which is the result of the LangSAM tool [22]. The segmentation of the product on the image can fail. Hence, either only a part of the product or no mask is segmented. Image examples are included in Section 10 in the Appendix. Moreover, the impact of the embedding model and the *Vector Store* type can be investigated in more depth. Further, reducing the context in term of the number of samples enables a valid response from the VLM used. Hence, the reduction of the context in relation to the image resolution can be investigated. Consequently, the printed text on images may then be no longer legible. In addition, novel rapidly emerging VLMs and new findings in term of Prompt Engineering can be analyzed. The analysis of different VLMs used in the Visual RAG pipeline shows that the model has significant influence of the result quality. This also means that Visual RAG pipeline is limited due to the external VLMs. The significant advantage of the presented Visual RAG pipeline is the minimal effort for new/unknown retail products. In contrast to the training of an image/text classifier, only few-shot samples of such product has to be provided in the *Vector Store*. In Section 5, the error analysis demonstrates the limitations. Numerous false predictions for the target *GTINs* are based to the insufficient data quality. Therefore, data cleaning, especially focusing on the product data *GTINs*, is essential. Consequently, several more investigations on the dataset and the Visual RAG pipeline may endure.

## References

- [1] Mohammad Mahdi Abootorabi, Amirhosein Zobeiri, Mahdi Dehghani, Mohammadali Mohammadkhani, Bardia Mohammadi, Omid Ghahroodi, Mahdiah Soleymani Baghshah, and Ehsaneddin Asgari. Ask in any modality: A comprehensive survey on multimodal retrieval-augmented generation, 2025. 1, 2
- [2] GS1 AISBL. Global Trade Item Number (GTIN). [https://www.gs1.org/docs/idkeys/GS1\\_GTIN\\_Executive\\_Summary.pdf](https://www.gs1.org/docs/idkeys/GS1_GTIN_Executive_Summary.pdf), 2015. [Accessed 01-03-2025]. 1
- [3] Amit Alfassy, Assaf Arbelle, Oshri Halimi, Sivan Harary, Roei Herzig, Eli Schwartz, Rameswar Panda, Michele Dolfi, Christoph Auer, Peter Staar, Kate Saenko, Rogerio Feris, and Leonid Karlinsky. Feta: Towards specializing foundational models for expert task applications. In *Advances in Neural Information Processing Systems*, pages 29873–29888. Curran Associates, Inc., 2022. 2
- [4] Christoph Brosch, Alexander Bouwens, Swen Haab, Sebastian Bast, and Rolf Krieger. Creation and evaluation of a food product image dataset for product property extraction. In *Proceedings of the 12th International Conference on Data Science, Technology and Applications (DATA)*, pages 488–495, 2023. 2
- [5] Zhanpeng Chen, Chengjin Xu, Yiyan Qi, and Jian Guo. Mllm is a strong reranker: Advancing multimodal retrieval-augmented generation via knowledge-enhanced reranking and noise-injected training, 2024. 2
- [6] Samuel Colvin, Marcelo Trylesinski, Sydney Runkle, Adrian Garcia Badaracco, Alex Hall, Hasan Ramezani, David Hewitt, and David Montague. Welcome to Pydantic - Pydantic — docs.pydantic.dev. <https://docs.pydantic.dev/latest/>. [Accessed 05-03-2025]. 5
- [7] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021. 1
- [8] Hugging Face. Hugging face – the ai community building the future. <https://huggingface.co>. [Accessed 01-03-2025]. 6
- [9] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Haofen Wang, and Haofen Wang. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*, 2, 2023. 1
- [10] Google. Gemini models. <https://ai.google.dev/gemini-api/docs/models/gemini>. [Accessed 27-02-2025]. 1, 5, 7
- [11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 3, 4, 5, 6
- [12] Samuel Hoffstaetter, Juarez Bochi, Matthias Lee, Lars Kistner, Ryan Mitchell, Emilio Cecchini, John Hagen, Dariusz Morawiec, Eddie Bedada, and Uğurcan Akyüz. Python tesseract. <https://github.com/madmaze/pytesseract>. [Accessed 01-03-2025]. 5, 1
- [13] Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. Openclip, 2021. 4
- [14] Chroma Inc. Chroma is the open-source ai application database. batteries included. <https://www.trychroma.com/home>. [Accessed 01-03-2025]. 4
- [15] Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of naacL-HLT*. Minneapolis, Minnesota, 2019. 3, 5, 6
- [16] Marcus Klasson, Cheng Zhang, and Hedvig Kjellström. A hierarchical grocery store image dataset with visual and semantic labels. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2019. 2
- [17] Jonathan Krause, Jia Deng, Michael Stark, and Li Fei-Fei. Collecting a large-scale dataset of fine-grained cars. 2013. 2
- [18] Bianca Lamm and Janis Keuper. Retail-786k: a large-scale dataset for visual entity matching, 2024. 2
- [19] Bianca Lamm and Janis Keuper. Can visual language models replace ocr-based visual question answering pipelines in production? a case study in retail, 2024. 5
- [20] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474, 2020. 2
- [21] Siyong Liu and Yili Zhao. Yub-200: A dataset for fine-grained bird recognition. In *2024 7th International Conference on Pattern Recognition and Artificial Intelligence (PRAI)*, pages 259–263. IEEE, 2024. 2
- [22] Luca Medeiros. Language segment-anything. <https://github.com/luca-medeiros/lang-segment-anything>. [Accessed 01-03-2025]. 4, 5, 8
- [23] OpenAI OpCo. Models. <https://platform.openai.com/docs/models>. [Accessed 01-03-2025]. 1, 4, 5, 7
- [24] OpenAI OpCo. GPT-4o mini: advancing cost-efficient intelligence. <https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence/>, 2024. [Accessed 01-03-2025]. 1, 5, 6, 7
- [25] Michelle Pokrass. Introducing Structured Outputs in the API. <https://openai.com/index/introducing-structured-outputs-in-the-api/>. [Accessed 01-03-2025]. 5
- [26] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 6

- [27] Varun Nagaraj Rao, Siddharth Choudhary, Aditya Deshpande, Ravi Kumar Satzoda, and Srikar Appalaraju. Raven: Multitask retrieval augmented vision-language learning, 2024. [2](#)
- [28] Xiao Wang, Guangyao Chen, Guangwu Qian, Pengcheng Gao, Xiao-Yong Wei, Yaowei Wang, Yonghong Tian, and Wen Gao. Large-scale multi-modal pre-trained models: A comprehensive survey. *Machine Intelligence Research*, 20(4):447–482, 2023. [1](#)
- [29] Xiu-Shen Wei, Yi-Zhe Song, Oisín Mac Aodha, Jianxin Wu, Yuxin Peng, Jinhui Tang, Jian Yang, and Serge Belongie. Fine-grained image analysis with deep learning: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 44(12):8927–8948, 2021. [1](#), [2](#)
- [30] Michihiro Yasunaga, Armen Aghajanyan, Weijia Shi, Rich James, Jure Leskovec, Percy Liang, Mike Lewis, Luke Zettlemoyer, and Wen tau Yih. Retrieval-augmented multimodal language modeling, 2023. [2](#)
- [31] Shi Yu, Chaoyue Tang, Bokai Xu, Junbo Cui, Junhao Ran, Yukun Yan, Zhenghao Liu, Shuo Wang, Xu Han, Zhiyuan Liu, and Maosong Sun. Visrag: Vision-based retrieval-augmented generation on multi-modality documents, 2024. [2](#)
- [32] Min Yuan, Ningning Lv, Yongkang Dong, Xiaowen Hu, Fuxiang Lu, Kun Zhan, Jiacheng Shen, Xiaolin Wu, Liye Zhu, and Yufei Xie. A dataset for fine-grained seed recognition. *Scientific Data*, 11(1):344, 2024. [2](#)
- [33] Jingyi Zhang, Jiaying Huang, Sheng Jin, and Shijian Lu. Vision-language models for vision tasks: A survey, 2024. [1](#)
- [34] Yi Zheng, Qitong Wang, and Margrit Betke. Semantic-based sentence recognition in images using bimodal deep learning. In *2021 IEEE International Conference on Image Processing (ICIP)*, pages 2753–2757. IEEE, 2021. [2](#)