https://doi.org/10.2352/EI.2025.37.8.IMAGE-262

Automating Product Image Analysis for Retail with Gemini

Tianli Yu, Daniel Vlasic; Google Cloud AI; Sunnyvale, California, USA

Abstract

We present the application of a Multimodal Large Language Model, specifically Gemini [1], in automating product image analysis for the retail industry. We demonstrate how Gemini's ability to generate text based on mixed image-text prompts enables two key applications: 1) Product Attribute Extraction, where various attributes of a product in an image can be extracted using open or closed vocabularies and used for any downstream analytics by the retailers, and 2) Product Recognition, where a product in a user-provided image is identified, and its corresponding product information is retrieved from a retailer's search index to be returned to the user.

In both cases, Gemini acts as a powerful and easily customizable recognition engine, simplifying the processing pipeline for retailers' developer teams. Traditionally, these tasks required multiple models (object detection, OCR, attributes classification, embedding, etc) working together, as well as extensive custom data collection and domain expertise. However, with Gemini, these tasks are streamlined by writing a set of prompts and straightforward logic to connect their outputs.

Introduction

Product images are a goldmine of information for online retailers, but this information remains largely under-utilized due to its unstructured nature. Effective utilization in search, recommendation, and analytics systems requires assigning relevant tags or attributes to each image. Traditionally, this involves specialized computer vision models coupled with laborious human verification to ensure accuracy. Building and maintaining these complex systems demands significant engineering resources and is hard to adapt to the dynamic nature of retail trends.

Large Language Models (LLMs), with their extensive world knowledge and multimodal capabilities, offer a transformative solution (Figure 1) in this domain. By leveraging an LLM like Gemini, product attribute extraction, for example, can be dramatically simplified. Instead of intricate model training, concise prompts can effectively guide the LLM to identify and extract desired information. This agility allows for rapid adaptation to evolving requirements and effortless expansion across diverse product categories, eliminating the need for model retraining.

In this paper, we delve into Gemini's capabilities for automated product image analysis. We explore two real-world use cases – product attribute extraction and product image recognition – demonstrating how this technology could empower retailers to unlock the full potential of their product images.

Related Work

This work lies at the intersection of computer vision, natural language processing, and information retrieval, drawing inspiration from advancements in Multimodal Large Language Models (MLLMs) and their applications in the retail domain.

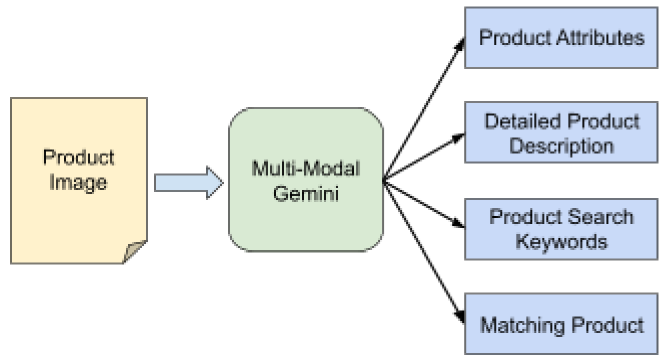

Figure 1. *Multimodal Gemini can parse a product image into a wide variety of information useful in the retail industry.*

Multi-modal Learning: The emergence of MLLMs like Gemini builds upon significant progress in multimodal learning, where models learn to process and relate information from different modalities, such as images and text. Early works focused on joint representations for image captioning [2, 3] and visual question answering [4, 5]. More recently, transformer-based architectures [6, 7] have enabled significant improvements in these tasks by effectively capturing cross-modal interactions.

Large Language Models for Vision: LLMs have demonstrated remarkable capabilities in understanding and generating human language [8, 9]. Recent research has explored extending these capabilities to the visual domain. CLIP [10] learns visual concepts from natural language supervision, enabling zero-shot image classification. ALIGN [11] aligns image and text representations at scale, demonstrating strong performance in image retrieval and understanding. These works pave the way for MLLMs like Gemini to perform complex visual reasoning tasks through natural language interfaces.

Product Image Analysis: Traditional approaches to product image analysis heavily rely on computer vision techniques, including object detection [12, 13], image classification [14, 15], and optical character recognition (OCR) [16, 17]. These methods often require extensive training data and specialized models for different tasks and product categories. Efforts have been made to leverage deep learning for attribute extraction [18, 19] and product recognition [20], but these approaches often suffer from limited generalization and require significant engineering efforts.

LLMs in Retail: LLMs have shown promise in various retail applications, including product description generation from text input [21], customer service [22], and personalized recommendations [23]. However, their application in automated product image analysis remains relatively unexplored. This work aims to address this gap by demonstrating the effectiveness of Gemini in extracting product attributes and facilitating accurate product recognition.

Product Attribute Extraction

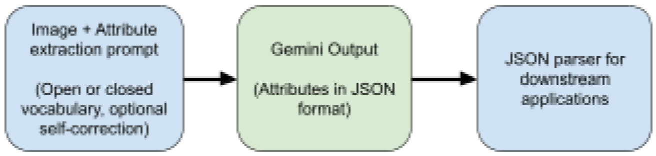

***Figure 2**. Gemini prompting flow for product attribute extraction.*

Product attribute extraction is a task that extracts text attributes (or tags) from a product image. These tags are useful to create different "drill-down" filters in a retailer's search interface, or they can be used as features in a product recommendation system. These attributes and their values can come from a pre-defined list of terms (closed vocabulary), or it can contain freeform text (open vocabulary). As mentioned in the introduction, traditionally this task needs to be solved using multiple different models (object detection, segmentation, and attribute classification). With multimodal foundational models like Gemini, all these can be simplified into a single call to the model with a multi-step prompt. The overall prompting flow for product attribute extraction is illustrated in Figure 2.[1]

Open Vocabulary Attribute Extraction

Prompt 1. Base product attribute extraction

> [System instructions] As an assistant for an online retailer, your task is to recognize attributes from the provided product image. If an attribute vocabulary is provided, please only select attribute values in that vocabulary. Your answer should be strictly consistent with what's in the image. If any attributes do not exist in the image, please return null for that attribute.
>
> {Input Product Image}
>
> [Detailed instructions] The above image is a product image from the {product_category} category. Please list all the relevant attributes in the {product_category} category for the main product in the above image and return a list of key-value pairs in json format.

Under Prompt 1, Gemini uses its world knowledge to automatically decide what attributes are relevant to the product category and extract their values. The extracted product attributes for a pair of women' s shoes are shown in Figure 3.

Self-Correcting Prompt

In Figure 3, we see that the extracted attributes have an obvious error: "toe style": "toe". In other cases, the model might generate an attribute that does not match the object in the picture. This is the well-known hallucination problem of LLMs that can be addressed with self reflection [24]. This simple technique uses a self-correcting prompt to instruct Gemini to check its answers and correct mistakes, effectively acting as a specialized model trained for the correction task.

```
{
  "upper material": ["genuine leather"],
  "heel style": ["stiletto"],
  "toe style": ["toe"],
  "shoe style": ["formal"],
  "color": ["red"]
}
```

***Figure 3**. Input image and extracted open vocabulary attributes.*

Prompt 2. Self-correcting product attribute extraction

> Next, treat the returned json as the result generated by a different model, rate each key-value pair as "correct" or "wrong" based on the same image. You can output in a format like "key - value: correct (or wrong)".
>
> Then, based on this evaluation, please update all the attributes that are corrected in the final json output. Please use markdown to annotate different json in your output.

If we add Prompt 2 to Prompt 1 and regenerate, the output of Gemini is shown in Figure 4. We noticed that Gemini generated more attributes with this prompt, and at the same time it corrected the two errors it made in the initial generation.

````
```json
{"gender": ["female"], "shoe style": ["formal"], "heel
style": ["stiletto"], "toe style": ["toe"], "color":
["red"], "occasion": ["null"], "material": ["genuine
leather"], "upper material": ["genuine leather"], "shoe
width": ["medium"], "arch support": ["high arches"],
"heel height": ["high heel"]}
```

gender - female: correct
shoe style - formal: correct
heel style - stiletto: correct
toe style - toe: wrong
color - red: correct
occasion - null: correct
material - genuine leather: correct
upper material - genuine leather: correct
shoe width - medium: correct
arch support - high arches: wrong
heel height - high heel: correct
```json
{"gender": ["female"], "shoe style": ["formal"], "heel
style": ["stiletto"], "toe style": ["peep toe"],
"color": ["red"], "occasion": ["null"], "material":
["genuine leather"], "upper material": ["genuine
leather"], "shoe width": ["medium"], "arch support":
["null"], "heel height": ["high heel"]}
```
````

Figure 4. Gemini response for the self-correcting prompt showing the corrected mistakes (bold text) in its initial result.

Closed Vocabulary Attribute Extraction

For closed vocabulary attribute extraction, we extend the base Prompt 1 with Prompt 3, which contains a dictionary of desired attributes and their possible values. This will ensure that Gemini generates an output expected by the user.

[1] A Google Colab notebook including all the prompts for product attribute extraction is available at: https://github.com/GoogleCloudPlatform/generative-ai/blob/main/gemini/use-cases/retail/product_attributes_extraction.ipynb

Prompt 3. Closed vocabulary attribute extraction

> Please use only the vocabulary defined in the following json:
>
> {Vocabulary JSON}
>
> For each key, you should select the most appropriate attribute value from its corresponding vocabulary list and return one value for each attribute key. You can return null for that key if no attributes match.

Figure 5 shows the closed vocabulary and the corresponding extracted attributes for the input image from Figure 3.

Closed vocabulary for the shoe category:

```
{
  "Pattern": ["Animal", "Letter", "Plaid", "Plain",
"Polka Dot", "Quilted", "Striped", "Tie Dye",
"Tropical", "Zebra", "Block", "Rainbow", "Floral"],
 "Toe": ["Almond Toe", "Cap Toe", "Closed Toe", "Peep
Toe", "Point Toe", "Pointed Toe", "Round Toe", "Square
Toe", "Toe Post", "Open Toe"],
 "Style": ["Ballet", "Bandage", "Basics", "Casual",
"Classic", "Cute", "Elegant", "Formal", "Modern",
"Motorcycle", "Retro", "Sexy",  "Boho", "Modest",
"Comfort", "Minimalist"],
 "Strap Type": ["Adjustable", "Ankle cuff", "Ankle
straps", "Chain", "Convertible", "Criss Cross",
"D'orsay", "Double Handle", "Flowers", "Gladiator",
"Lace Up", "Mary Jane", "Ring", "Slingbacks",
"Strappy", "T strap", "Zipper", "Elastic", "Velcro",
"Ankle Strap"],
 "Heels": ["Chunky", "Cork", "Espadrilles", "Flat",
"Flatform", "Platform", "Stiletto", "Cone Heel",
"Kitten Heels", "Hidden Wedge", "Wedges", "Pyramid"],
 "Closure Type": ["Back Zipper", "Buckle", "Zipper",
"Magnet", "Slip on", "Hook Loop", "Lace-up", "Flap"]
}
```

Extracted attributes:

```
{
  "Pattern": ["Plain"],
  "Toe": ["Peep Toe"],
  "Style": ["Sexy"],
  "Strap Type": ["null"],
  "Heels": ["Stiletto"],
  "Closure Type": ["Slip on"]
}
```

Figure 5. *Closed vocabulary for the shoe category and the resulting attributes.*

Product Image Recognition

In addition to extracting fine-grained attributes from a product image, the Multimodal Gemini is also an excellent product recognizer. When paired with a text search engine, Gemini enables precise product recognition functionality from any user input image. Usually, image-based product search is designed to employ image embedding models in combination with object detection or segmentation models. With Gemini's excellent recognition and text generation capabilities, we can achieve the same functionality by making a pair of Gemini adapters to a text search engine. The ability to work with an existing text search engine is very important as retailers can leverage the search engine they already built for text search and save time and cost.

Figure 6 shows the processing flow of Gemini-adapted product image recognition. Multimodal capabilities allow Gemini to convert any image to a text description or keywords, which can then be fed into the existing text search engine. The text search results can also be processed by Gemini to select the exact product matching the one in the user-provided image.

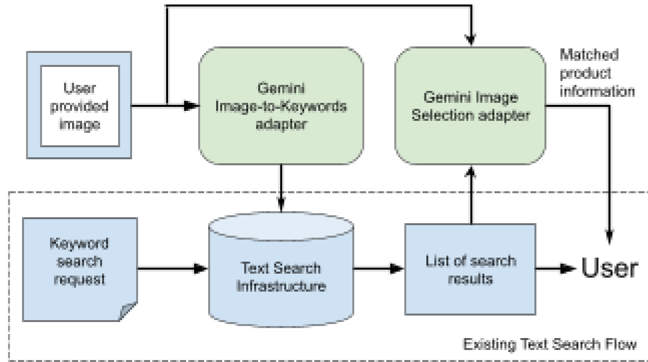

Figure 6. Gemini product recognition flow diagram.

Search Keywords Generation

Prompt 4. Search keywords generation from an image

> {Input Image}
>
> If I want to search a retailer's website for the product in this image, what keywords should I use? Please output only the keywords. In general, you should use a format like brand, product name, product variant. If there are multiple products in the image, you should only select the most prominent one.

In the first stage of Gemini-powered product recognition, the prompt (Prompt 4) focuses on how to extract concise and accurate keywords to support the downstream text search stage.

 "Orijen Original Cat"	 "Greenies Savory Salmon Flavor Dental Treats"	 "Sue Bee, Pure Honey"
 "JUST Egg, Plant-based scramble"	 "Apple & Eve, 100% Juice, 36 pack"	 "Skippy, Creamy Peanut Butter, Single Serve Pouch"
 "white bookshelf, 5 shelf bookshelf"	 "Crayola, Crayons, 24 count"	 "Kellogg's, Corn Flakes, Original & Best"

Figure 7. *Keywords generated by Gemini using Prompt 4.*

Figure 7 shows examples of keywords generated from product images using Prompt 4. Note that with a multimodal LLM, not only can one extract the text describing the product in the image, but also perform content summarization as shown in the top right image. Even though an image is a back of the product container with many words on it, Gemini filters out many detailed words and only keeps the ones that are useful to identify the product.

Matching Product Selection

Prompt 5. Matching product selection

> You are a product search engine and your responsibility is to recognize the product in the query image by selecting the closest image from a set of candidates. Please follow these instructions:
> a. The overall picture layout and content of the candidate should match exactly with the product in the query image.
> b. The title of the candidate should also match the product in the query image.
> c. Pay attention to details on the product label like product name, product variant, product weight, size and quantity.
> d. You should also pay attention to the package type, shape and color scheme which could indicate different variants, weight and quantity.
> e. The query images are captured in non-ideal conditions and might have distortions or occlusions. If the image is blurry or incomplete, you should find the best match based on partial similarity of graphics and text layout on the packaging.
>
> Please output the index of the closest matched candidate only without any markup or explanation.
>
> Query Image: {Input Image}
> Candidate 0: {candidate 0 image}
> Candidate 1: {candidate 1 image}
> Candidate 2: {candidate 2 image}
> …

The generated keywords can be used to search for the product using an existing text search engine. The text search engine might return a list of search results, where the top one is not necessarily the best match. For this case, we designed a second Gemini prompt (Prompt 5) to select the best-matching product from a list of search results (usually the top 10 results are enough).

Figure 8. *Top: Query image (left) and matching product selected by Gemini (right). Bottom: candidate search results returned by the text search engine.*

Figure 8 shows an example of matching product selection by Gemini. Due to the fact that some products have many variants, it's hard for keyword search to rank the matching products. In this case Gemini can be used to precisely locate the matching product.

Experiments

Product Attribute Extraction

We evaluate the product attribute extraction using a dataset with 5 different categories from a single online retailer. The online retailer's performance expectation is 90% accuracy on this test set based on their previous experience and their non-LLM system's overall performance. The attributes are extracted by Gemini Pro under a closed vocabulary setting and the extracted attributes are rated by a human rater for correctness. If an attribute is not relevant for a particular product, it is expected to be "null" and a non-null answer is treated as an error.

Table 1. Accuracy for closed vocabulary product attribute extraction with and without self-correcting prompt.

	Shoes	Bags	Nails & Wigs	Phone Cases	Underwear
# of eval images	29	29	29	29	33
# of attributes	7	10	5	5	17
Accuracy % before self-correction	88.7	91.7	**93.1**	92.4	92.9
Accuracy % after self-correction	**94.0**	**94.1**	**93.1**	**93.8**	**97.8**
Total # of self-corrections	11	13	9	18	42

From Table 1, we can see that Gemini's performance has clearly exceeded the retailer's requirements. The ablation study on the effectiveness of self-corrections shows that the self-correcting prompt can significantly boost the extraction accuracy and achieve the best results on this dataset. Note that self-correction does not always produce a better result, as shown in the Nails and Wigs category. There were 9 self-corrections but the accuracy before and after self-corrections stays the same.

Analyzing the remaining errors, we found that, although Gemini did an overall great job extracting clear attributes from the images, it does struggle in some ambiguous cases. Figure 9 shows one typical error case, where the ankle bracelet on the model is mistakenly recognized as the "ankle strap" of the shoes. This means that Gemini currently does not have enough reasoning capabilities to differentiate such cases.

Figure 9. *Error cases of Gemini for shoe attribute extraction: ankle bracelet being recognized as "ankle strap".*

Product Image Recognition

The performance of the product image recognition flow depends on three things: 1. the quality of generated keywords; 2. the retrieval quality given these keywords; 3. the final matching product selection quality. The first two link to various external factors like retrieval index size and the performance of the retrieval engine, while the third can be quantified independently. Therefore, our evaluation will focus on the quality of the final matching product selection of Gemini.

We evaluated the matching quality on two internal retail product image datasets. The images in these datasets are captured from several retailer's shelves. Figure 10 shows some examples of the query image.

Figure 10. *Typical query images captured from store shelves in the evaluation datasets.*

We paired these images with two different retrieval systems (A and B), where retrieval system A has a smaller index size than system B. From the second row of Table 2, we can see that the retrieval performance of System A is lower than System B. Note that the first row is also the upper bound of the product recognition accuracy in the subsequent product matching step using Gemini.

Table 2. Percentage of correctly recognized products by Gemini Pro and Flash.

	Dataset A /Retrieval System A	Dataset A /Retrieval System B	Dataset B /Retrieval System A	Dataset B /Retrieval System B
GT in retrieved results (Upper Bound)	93.35%	99.67%	80.0%	96.23%
Baseline – Conventional Model	77.66%	86.39%	73.40%	90.26%
Gemini 1.5 Flash	83.57%	91.91%	**76.34%**	**92.57%**
Gemini 1.5 Pro	**83.59%**	**92.33%**	75.92%	91.94%
Estimated Human Operator Performance	N/A	95 ~ 96%	N/A	93~94%

Table 2 shows that both Gemini 1.5 Pro and Flash significantly outperform the baseline, which is built using conventional AI models. We also estimated the human operator's performance on retrieval system B where the retrieval gap is relatively small. The human operator's performance is limited by certain image quality constraints such as the image is too blurred to identify the product.

The performance difference between Gemini 1.5 Pro and Flash is relatively small, which indicates that customers might want to take advantage of Gemini 1.5 Flash's low cost and high inference speed for many practical applications in the retail industry.

Discussion

In addition to accurate extraction of product attributes and product recognition from images, Multimodal LLMs have other benefits. One of the advantages is the flexibility, where the model's output can be easily changed using different prompts without the need for retraining. Another benefit is that it is very easy to add additional instructions to ask the LLM to explain its decisions. This makes it easier to understand the model behavior, and can help with debugging.

Conclusions

We show that recent developments in Multimodal LLMs enable novel image analytics applications in the retail space, from product attribute extraction to accurate product recognition. Multimodal LLMs provide a simple, easy-to-use, and training-free approach to building these applications. Furthermore, performance improvements in smaller models like Gemini Flash make this solution more cost effective. We believe that retailers will increasingly adopt these capabilities in their search and recommendation pipelines.

Acknowledgements

We'd like to thank Rui Lin, Jingya Liu, Michael Moynihan for their help with the product recognition experiments. We'd also like to thank Brendan Lundy for providing test images for the experiments in this paper. Finally, we thank Chao Jia, Ting Yu and Shengyang Dai for their advice and support for this project.

References

[1] Gemini Team Google, Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context, arXiv:2403.05530 [cs.CL]

[2] Oriol Vinyals; Alexander Toshev; Samy Bengio; Dumitru Erhan (2015). Show and tell: A neural image caption generator. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 3156-3164).

[3] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, Yoshua Bengio Show, attend and tell: Neural image caption generation with visual attention. Proceedings of the 32nd International Conference on Machine Learning, PMLR 37:2048-2057, 2015.

[4] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, Devi Parikh; VQA: Visual question answering. Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2015, pp. 2425-2433.

[5] Mateusz Malinowski, Marcus Rohrbach, Mario Fritz (2015). Ask your neurons: A neural-based approach to answering questions about images. In Proceedings of the IEEE international conference on computer vision (pp. 1-9).

[6] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, Illia Polosukhin

(2017). Attention is all you need. Advances in neural information processing systems, 30.

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. NAACL.

[8] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, Dario Amodei (2020). Language models are few-shot learners. Advances in neural information processing systems, 33, 1877-1901.

[9] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sashank Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, Noah Fiedel (2024). Palm: Scaling language modeling with pathways. The Journal of Machine Learning Research, Volume 24, Issue 1, Article No.: 240, Pages 11324 - 11436

[10] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, Ilya Sutskever (2021). Learning transferable visual models from natural language supervision. Learning Transferable Visual Models From Natural Language Supervision. ICML 2021: 8748-8763

[11] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V. Le, Yun-Hsuan Sung, Zhen Li, Tom Duerig, Scaling Up Visual and Vision-Language Representation Learning With Noisy Text Supervision. ICML 2021: 4904-4916

[12] Joseph Redmon; Santosh Divvala; Ross Girshick; Ali Farhadi (2016). You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 779-788).

[13] Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems, 28.

[14] Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton (2012). Imagenet classification with deep convolutional neural networks. Advances in Neural Information Processing Systems 25 (NIPS 2012).

[15] Karen Simonyan, Andrew Zisserman. (2015). Very Deep Convolutional Networks for Large-Scale Image Recognition. ICLR 2015.

[16] Ray Smith (2007). An overview of the Tesseract OCR engine. In Ninth international conference on document analysis and recognition (ICDAR 2007) (Vol. 2, pp. 629-633). IEEE.

[17] OCR with Google Cloud AI, https://cloud.google.com/use-cases/ocr (2024),

[18] Ziwei Liu, Ping Luo, Shi Qiu, Xiaogang Wang, Xiaoou Tang (2016). Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2016 (pp. 1096-1104).

[19] Junshi Huang, Rogerio Feris, Qiang Chen, Shuicheng Yan (2016). Cross-domain image retrieval with a dual attribute-aware ranking network. 2015 IEEE International Conference on Computer Vision (ICCV) (pp. 1062-1070).

[20] Sean Bell, Kavita Bala (2015). Learning visual similarity for product design with convolutional neural networks. ACM Transactions on Graphics (TOG), 34(4), 1-10.

[21] Chester Palen-Michel, Ruixiang Wang, Yipeng Zhang, David Yu, Canran Xu, Zhe Wu (2024). Investigating LLM Applications in E-Commerce. arXiv:2408.12779 [cs.CL].

[22] Jing Wei, Sungdong Kim, Hyunhoon Jung, Young-Ho Kim. (2022). Leveraging Large Language Models to Power Chatbots for Collecting User Self-Reported Data. Proceedings of the ACM on Human-Computer Interaction, Volume 8, Issue CSCW1, Article No.: 87, Pages 1 - 35.

[23] Wang-Cheng Kang, Julian McAuley (2018). Self-attentive sequential recommendation. In 2018 IEEE International Conference on Data Mining (ICDM) (pp. 197-206). IEEE.

[24] Ziwei Ji, Tiezheng Yu, Yan Xu, Nayeon Lee, Etsuko Ishii, and Pascale Fung. 2023. Towards Mitigating LLM Hallucination via Self Reflection. In Findings of the Association for Computational Linguistics: EMNLP 2023, pages 1827–1843, Singapore. Association for Computational Linguistics.

Author Biography

Tianli Yu received his M.Eng from Tsinghua University and Ph.D. in ECE from University of Illinois at Urbana-Champaign. He held several senior researcher and engineer positions in Motorola Labs, like.com and Google Shopping. From 2014-2020, He was also the co-founder and CEO of an educational robotics startup (Morpx). He's currently working in Google Cloud AI, focusing on the applications of Multimodal Gemini.

Daniel Vlasic received his BS, MEng, and PhD in Computer Science from Massachusetts Institute of Technology (2002, 2002, 2010). He has been working as a Software Engineer at Google since 2017, focusing on Computer Vision and Machine Learning research, as well as Large Multimodal Model quality.