MULTIMODAL CASCADED APPROACH FOR HIERARCHICAL LOGO TAGGING INPACKAGING ARTWORK FILES

Shishir Maurya, Anshul Verma, Yugal Gopal Sharma,
Dhanush Dharmaretnam

SGS&CO, Louisville, Kentucky, USA

ABSTRACT

This study proposes a novel method for recognizing and categorizing logos in packaging artwork to address the automation demands of the printing and packaging industry. The approach combines a trained object detection model for logo detection followed by a fine-tuned Vision Language Model (VLM) for hierarchical tag generation, achieving high precision across seven primary categories: sustainability, health and safety, branding, material identification, eco-friendly certification, social media, and compliance, with all others grouped under "others." In the first step, YOLOv8 detects logos and assigns them to primary categories, achieving a mean average precision (mAP) of 0.58 and an Intersection over Union (IoU) threshold of 0.5. In the second step, a fine-tuned VLM generates granular tags for the detected logos. Notably, Low Rank Adaptations (LoRA) applied to the Florence-2-DocVQA model (with r = 64 and α = 128) surpassed the zero-shot performance of state-of-the-art VLMs, achieving a 24-fold improvement with a ROUGE-L F1 score of 0.72. This study also demonstrates the cost effectiveness and practicality of using smaller models with fewer parameters, which perform comparably to larger VLMs, incurring much lower training and operational costs. These advancements streamline design and print production workflows, improve compliance tracking, and enhance brand management, contributing to greater automation in the packaging and printing industry.

KEYWORDS

Packaging Artwork, VLMs, Artwork Tagging, Low Rank Adaptation (LoRA)

1. INTRODUCTION

In the packaging industry, manual logo verification is prone to errors, leading to costly multi-million dollar recalls and highlighting the need for automated solutions. These manual processes cause delays and inconsistencies in brand identity, making them increasingly unsustainable due to the volume of packaging artwork that requires verification. Additionally, compliance with region-specific regulatory standards, such as the FDA's labelling requirements[1], adds complexity to manual verification.

Packaging artworks as shown in Figure 1, which integrate logos and other critical elements such as nutritional panels and barcodes, play a vital role in customer trust and informed purchasing. Errors in regulatory symbols can result in significant penalties, making accurate verification crucial for maintaining consumer confidence and avoiding legal repercussions. Thus, ensuring the accurate replication of branding and regulatory symbols remains a significant challenge.

David C. Wyld et al. (Eds): NLPCL, BIGML, AISC – 2025
pp. 65-88, 2025. CS & IT - CSCP 2025 DOI: 10.5121/csit.2025.151808

The proposed cascaded model addresses these challenges, delivering a 5-fold speedup in artwork verification compared to manual validation and reducing annotation costs by nearly 30% through an finetuned YOLOv8 model and a semi-supervised tagging approach using VLMs using custom private dataset representative of the industry usecase. This automation replaces the initial stage of human validation, ensuring faster, more accurate, and consistent brand identity checks.

While product packaging is critical for communicating information and maintaining brand consistency, current verification methods struggle to keep pace with the industry demands. Regional variations in manufacturing and distribution, coupled with the increasing complexity of packaging designs, often lead to inconsistencies and errors in brand representation. Errors in regulatory symbols can lead to significant penalties, making accurate and consistent verification crucial for maintaining consumer trust and avoiding legal repercussions. Hence, ensuring the accurate replication of branding and regulatory symbols remains a significant challenge, particularly given the subtle variations that can exist within logo families. Similar regulations worldwide govern layout, content clarity, and accuracy, further underscoring the importance of compliance in packaging design.

Automated logo detection helps minimize errors and protect brand integrity, but subtle variations, as illustrated in Figure 2, like changes in color, shape, or text, can lead to misclassifications. Cluttered packaging designs further complicate accurate logo detection, as shown in examples of subtle variations in sustainability logos that require context-based analysis for distinction.

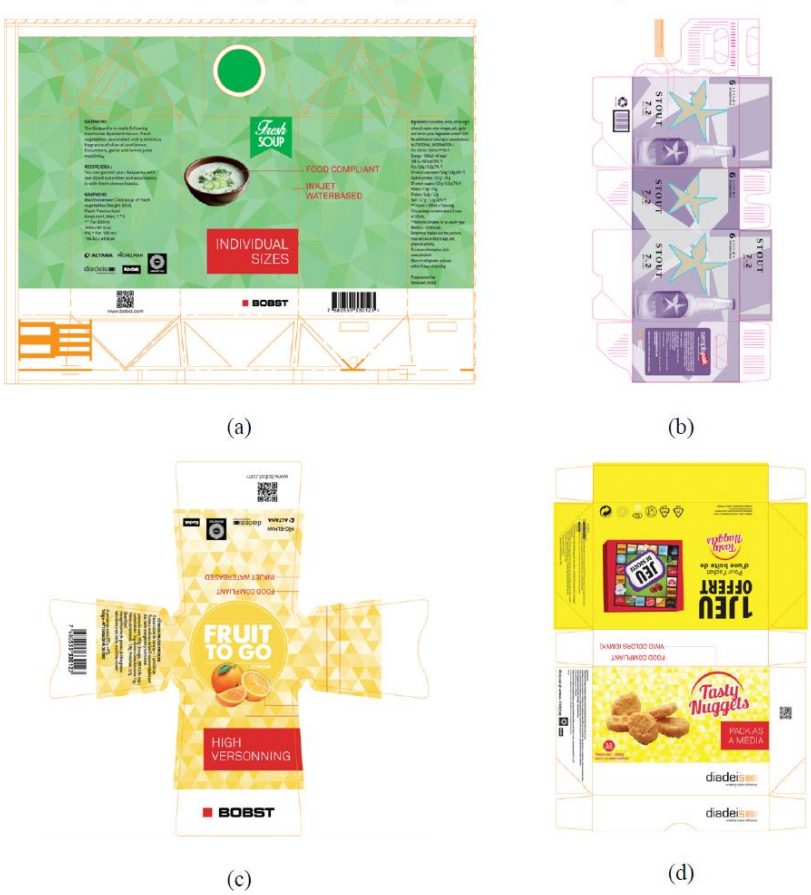

Figure 1. Packaging artworks from various products, featuring logos for branding, regulation, and sustainability.

1.1. Related Work

Automated logo detection in product packaging has advanced significantly with deep learning. Hou et al. [2] demonstrated the use of deep learning models to automate logo identification, moving away from manual methods. Su et al. [3] introduced Context Adversarial Learning for improved performance across different contexts, while Zhao et al. [4] presented DCFNet, achieving impressive results in small logo detection.

The use of synthetic datasets has lessened the need for manual annotations. Mas Montserrat et al. [5] applied synthetic methods for logo localization, and LOGO-Net [6] advanced brand recognition, though regulatory symbols are still not well-explored. Recent advancements in self-supervised and self-attention techniques [7] have further boosted detection accuracy in complex scenarios.

(a) Recycling Logos

(b) Sustainability Logos

Figure 2. Examples of visually similar logos: (a) Recycle logos indicating different materials and whether they can be recycled. (b) Sustainability logos representing various product attributes such as eco-friendly, cruelty-free, and vegan.

Challenges persist in logo classification due to numerous similar subclasses. Hybrid models combining object detection with context-based tagging are gaining traction. Brailovsky et al. [8] focused on logo differentiation in varied image types, while Hu et al. [9] enhanced CNN-based recognition with contextual information, emphasizing multimodal techniques.

Automated logo detection has shown progress [7, 10], but despite these advancements, regulatory symbols crucial to compliance remain underexplored. Specialized models are needed to tackle complex designs and ensure accurate and efficient processing of packaging artwork.

1.2. Research Contribution

This paper introduces a cascaded framework that combines object detection and VLMs for identifying and classifying logos in packaging artwork. The main contributions of this study are:

1. A fast and scalable logo detection system for packaging artworks, addressing various logo types, highlighting the brand, health and safety, regulatory and compliance symbols,
2. A low-cost, semi-supervised tagging approach for effective logo categorization, and
3. A demonstration of significant time and cost savings in real-world packaging workflows, enhancing efficiency and accuracy.

By transitioning from manual artwork quality checks to a semi-automated, model-driven process, the proposed method improves operational efficiency and ensures consistent brand identity across diverse packaging designs.

2. METHODOLOGY

This study employs a two-stage, cascaded approach for detecting and tagging logos in packaging artwork images. The first stage uses object detection to identify logos and assign them primary categories, while the second stage applies a tagging phase to generate more granular classifications through label assignment. This dual-stage design enhances classification accuracy for logos with subtle variations.

2.1. Dataset Preparation

2.1.1. Logo Detection Dataset

The logo detection dataset was created using dummy artwork files, ensuring that no real client data was used. A human-in-the-loop (HITL) approach was adopted for manual labelling and validation.Unlike conventional logo detection datasets, such as the LogoDet-3k dataset [11] and the Open Logo Detection Challenge dataset[3], which primarily feature brand logos extracted from product packaging, advertisements, or real-world scenes, the dataset used in this study is uniquely curated for artwork contexts. It captures a broader range of logos beyond typical brand marks, including diverse categories such as sustainability, compliance, material identification, health and safety, eco-friendly certifications, and social media. This comprehensive approach ensures a more realistic representation of the varied logos encountered in artwork files, addressing the gap left by traditional datasets.

Initially, a subset of 50 artwork images was randomly selected from the artwork database. The logos in these images were manually labelled with bounding box coordinates and assigned primary classes and were used for fine-tuning the YOLOv8 model using COCO-pretrained weights. The fine-tuned model was then used to infer logos on unannotated images, and these detections were manually validated and corrected. The corrected detections were added to the training set, and this iterative process continued until the YOLOv8 model achieved a mAP threshold of 0.8 across all classes. This resulted in a final dataset of 432 annotated images spanning seven primary classes: branding, sustainability, materialidentification, health and safety, compliance, eco-friendly certifications, socialmedia, and an “others” category.

The dataset was split into training (70%), validation (20%), and test (10%) sets. The class distribution for the entire dataset is shown in Figure 3.The workflow of the logo detection data annotation is shown in Figure 4. This method was used to create the logo detection dataset to train the first stage of the cascaded approach.

2.1.2. Logo Tagging Dataset

The logo tagging dataset was created using logos present in dummy artwork files. Initially, large language models (LLMs) were used to suggest potential tags for manually labelled logos, which were then validated and corrected by human annotators to ensure accuracy. Each logo was tagged with a primary class followed by additional descriptors representing its subcategory.

For initial tagging, the VLM, *Llama-3.2-11B-Vision-Instruct*, was used to generate suggested tags through a structured prompt. The generated tags were stored in a JSON file as key-value pairs, where logos are keys and the corresponding tags are values for training the tagging model. Figure 5 depicts the process of creating the tagging dataset using the HITL and VLM-assisted approach.

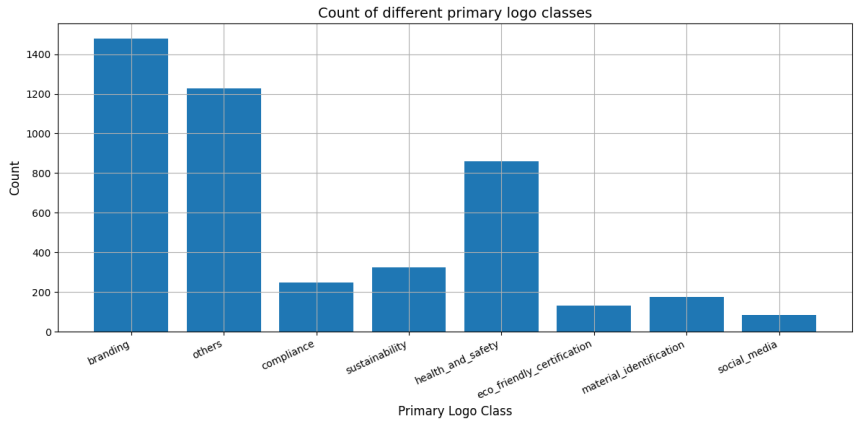

Figure 3. Distribution of primary logo classes in thefinal training dataset.

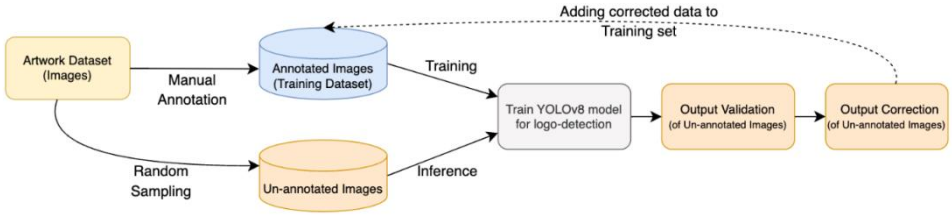

Figure 4. Workflow of logo detection data annotation using YOLOv8.

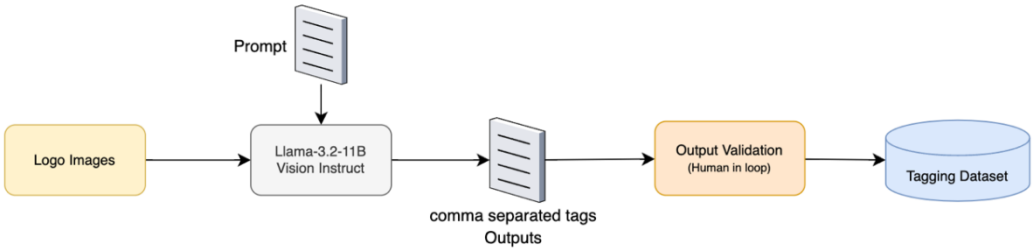

Figure5. Workflow of tagging dataset creation using the Llama-3.2-11B-Vision-Instruct model.

It is to be noted that only logos detected in the artwork images by the first step of the cascaded approach were included in the tagging dataset. Each detected logo was paired with its primary

class, and the VLM generated relevant descriptive tags based on the visual features and class of the logo. These tags were then manually validated to ensure consistency across the dataset.

The final tagging dataset comprises 1,114 tagged logos, split into training, test, and validation sets in a 60:20:20 ratio. A sample of the tagging dataset is shown in Figure 6.

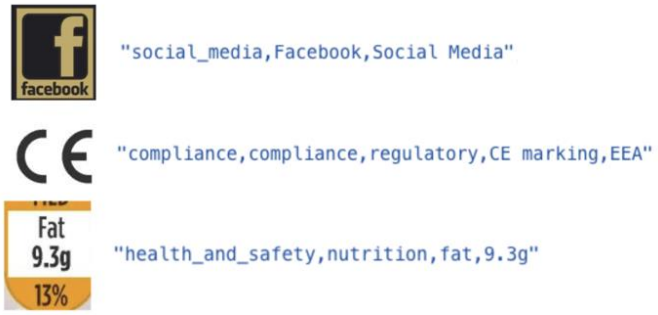

Figure 6. Sample logos and their tags in hierarchy alongside the primary class.

Figure 7shows the prompt used to generate the initial tags to prepare the logo tagging dataset, which was reviewed and corrected by an expert through the HITL approach. Moreover, this prompt was also used as a base prompt to compare the zero-shot tag generation performance of the VLMs such as Llama-3.2-11B-Vision-Instruct, GPT4o, Haiku, and Sonnet. On the other hand, the Florence-2 is a very light model and only accepts specified task prompts ("DETAILED_CAPTION" used in this case) along with high-level user prompts with small token lengths ("Generate comma-separated tags for the given logo image" in this case). This explains the superior zero-shot performance of GPT-4o, Haiku, Sonnet, and Llama-3.2 over Florence-2. On the other hand, "Generate comma-separated tags for the given logo image" was used for finetuning both Llama-3.2-11B-Vision-Instruct and Florence-2 models.

```
You are a logo identification expert tasked with classifying logos in
an image into one of the following categories based on visual or
textual elements:

   • sustainability: eco-friendly symbols (e.g., recycling,
     biodegradable)
   • health_and_safety: health/warning symbols, safety standards
   • branding: brand/company logos or trademarks
   • material_identification: material type logos (e.g., plastic,
     glass, paper)
   • eco-friendly-certification: environmental certifications (e.g.,
     organic, biodegradable)
   • social_media: social media platform logos (e.g., Facebook,
     Instagram)
   • compliance: compliance standards logos (e.g., CE marking, RoHS)
   • others: logos not fitting the above categories

Classify the logo based on image content. Only one category per logo.
The output should be in JSON format:

Example:

{ "logo_category": "category identified", "explanation": "brief
explanation", "tags": ["comma-separated list of tags"] }


Only give a valid JSON output and nothing else. No additional
explanation is required apart from the JSON. Strictly start your
answer with {
```

Figure 7. Prompt used to generate tags with "Llama-3.2-11B-Vision-Instruct" for the detected logo images.Generated tags are then verified by an expert.

2.2. Overall Approach

The proposed methodology follows a twin-model cascaded process, as shown in Figure 8 (a). The process consists of:

1. Multi-class object detection for identifying logos and assigning primary classes.
2. A tagging phase that refines the classification by generating additional labels to capture finer-grained logo details.

Figures 8(b) illustrates the diverse logo styles present in artwork files. This hierarchical classification approach ensures that even minor visual or symbolic distinctions are accurately captured.

2.3. Logo Detection

For logo detection, YOLOv8, Faster R-CNN, and DETR were trained and evaluated, with YOLOv8 selected for its balance of speed, accuracy, and computational efficiency. The model was fine-tuned on a curated dataset of 432 labelled packaging artwork images, designed to represent real-world packaging scenarios while excluding any proprietary or sensitive data.

The YOLOv8 model was optimized using its default composite loss function, combining bounding-box regression, objectness, and classification loss. All the detection models were trained using the weighted Adam optimizer with a learning rate of 10^{-4}, a batch size of 8, and for 100 epochs.

To enhance generalization and address class imbalance, advanced data augmentation techniques such as CutMix[12], MixUp[13], Mosaic[14], random horizontal flips, color-jitters, cropping, warping and rotations were applied.

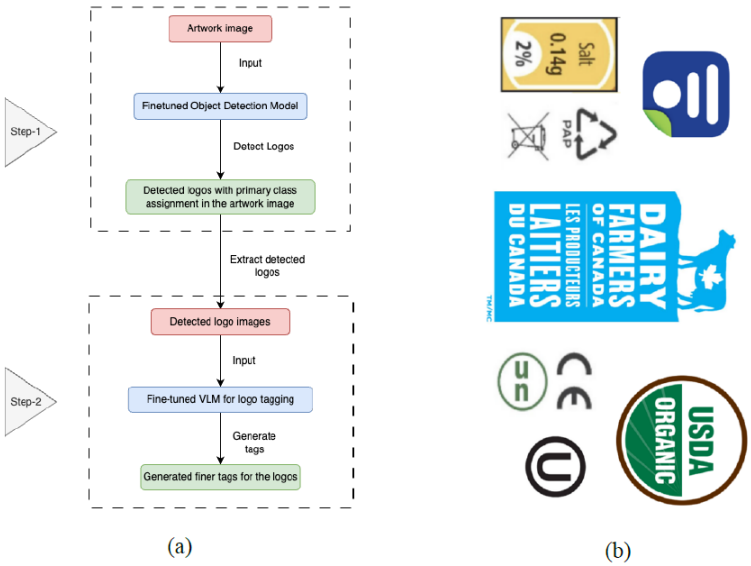

Figure 8. (a) Model block diagram of the cascaded approach to achieve logo detection and tagging. (b) Examples of logos from various packagingartworks, showcasing different designs and styles.

Training on the fully annotated dataset was conducted using an NVIDIA T4 GPU, leveraging efficient augmentation and batch size adjustments to reduce training time by approximately 20%.

Model performance for the detection phase was assessed using mAP scores at different IoU thresholds.

Logos in the artwork as shown in Figure 8 (b),often exhibit subtle differences that must be captured for accurate classification. These fine distinctions separate logos into primary and secondary classes. Other logotypes similarly vary in text, style, and the information they convey.

2.3. Logo Tagging

For the tagging task, the output from logo detection was processed using VLMs like GPT-4o[15],Llama-3.2[16], Haiku, Sonnet [17], and Florence-2 [18] in a zero-shot setting. Tags were generated based on the defined prompt shown in Figure 5.

Fine-tuning was performed on Llama-3.2-11B-Vision-Instruct and Florence-2-DocVQA using LoRA[19] and QLoRA[20]. Fine-tuning explored rank (r) and scaling factor (α) combinations, including (r=8, $\alpha = 16$), (r= 16, $\alpha = 32$), (r= 32, $\alpha = 64$), and (r= 64, $\alpha = 128$), for a total of 10 epochs. The masked language modelling loss was used, computing cross-entropy between predicted logits and true token labels. Training was performed on an A100 GPU with a batch size optimized for GPU utilization, using the Adam optimizer and a fixed learning rate of 10^{-4}.

3. RESULTS AND DISCUSSIONS

The proposed twin-model cascaded system was evaluated on both logo detection and tagging tasks, focusing on inference efficiency, training cost, and performance improvements.

3.1. Logo Detection

Object detection models were trained and evaluated at an IoU threshold of 0.5. Table 1 summarizes the performance. YOLOv8 achieved the highest mAP (0.578) with the fewest parameters, making it the most efficient in terms of both accuracy and inference cost. This model demonstrated optimal performance when deployed in a 2 GB CPU container, achieving an inference speed of 2.8 seconds per image—a balance of high latency and low computational cost.

Table 1. Experiments for logo detection at a confidence threshold of 0.5 on the test set, with **M** representing million. The best results are reflected in **bold**.

Base Model	**mAPIoU=0.50**	**mAPIoU=0.50:0.95**	**Number of parameters**
YOLOv8	**0.578**	0.361	**11.2 M**
Faster-RCNN	0.562	**0.406**	44 M
DETR	0.552	0.367	41 M

3.2. Logo Tagging Zero-Shot Performance

Tag generation was assessed using BLEU and ROUGE metrics in a zero-shot setting, as these are the standard evaluation metrics used to measure the quality of the generated text. As shown in Table 2, GPT-4o achieved the best results, outperforming models such as Llama-3.2-11B-Vision-Instruct and Florence-2-DocVQA, which struggled due to their limited task-specific capabilities. Notably, the larger models allow for more detailed prompting, enabling them to generate more specific outputs in a zero-shot setting. In contrast, Florence-2, being a lightweight model, only accepts specified task prompts (e.g., “DETAILED_CAPTION” in this case) and high-level user

prompts with limited token lengths. This constraint explains the superior performance of GPT-4o,Haiku,Sonnet, and Llama-3.2 over Florence-2 in zero-shot scenarios.

3.3. Logo Tagging LoRAFine-Tuning Performance

Fine-tuning experiments were conducted using the Llama-3.2-11B-Vision-Instruct and Florence-2-DocVQA models, applying both LoRA and 4-bit QLoRA configurations. Various scaling factors (α) and rank(r) were tested on the logo tagging dataset to evaluate the models' performance. Due to the significantly larger size of the Llama model compared to Florence-2, the former exhibited a lower ratio of trainable to total parameters, resulting in increased hallucinations after more epochs of training. The prompt for fine-tuning was also simplified from the original

Table2. *ZERO-SHOT* performance of VLMs for taggeneration on test set; best results are reflected in **bold**.

Model	**BLEU Score**	**ROUGE Score Type**	**P**	**R**	**F Score**
Llama-3.2-11B-Vision-Instruct	0.05	ROUGE-1	0.3	0.28	0.28
		ROUGE-2	0.13	0.12	0.11
		ROUGE-L	0.29	0.27	0.27
Florence-2-DocVQA	0	ROUGE-1	0.02	0.06	0.03
		ROUGE-2	0	0.01	0
		ROUGE-L	0.02	0.06	0.03
GPT-4o	**0.06**	ROUGE-1	**0.37**	**0.48**	**0.41**
		ROUGE-2	**0.14**	**0.22**	**0.17**
		ROUGE-L	**0.35**	**0.45**	**0.38**
Haiku	0.04	ROUGE-1	0.24	0.31	0.27
		ROUGE-2	0.07	0.1	0.08
		ROUGE-L	0.23	0.3	0.25
Sonnet	0.04	ROUGE-1	0.21	0.32	0.25
		ROUGE-2	0.06	0.09	0.07
		ROUGE-L	0.19	0.29	0.23

version (prompt in Figure 7) to a more direct instruction: "*Generate comma-separated tags for this logo.*"

In contrast, the Florence-2 model, with a higher ratio of trainable parameters to total parameters, demonstrated more stable fine-tuning performance. This allowed it to achieve satisfactory results with fewer parameters than the Llama-3.2 model. Fine-tuning performance, evaluated using ROUGE-1 F1 scores, is presented in Figure 9.

Moreover, Table 3 highlights the improvements in tagging performance with fine-tuning as the LoRA scaling factor (α) and rank (r) increase. Notably, Florence-2 outperformed Llama-3.2 during fine-tuning due to its larger proportion of trainable parameters. This enabled more efficient fine-tuning, improved context capture, and superior performance with fewer epochs. Despite being a smaller model, Florence-2 demonstrated significant adaptability to fine-tuning conditions, underscoring the cost-effectiveness of tuning smaller models with targeted training data.

The training cost for the fine-tuned LoRA Florence-2 model is approximately $1.50, while the fine-tuned Llama-3.2 model costs around $4.50. The training time is considerably reduced due to the smaller training dataset used. A fine-tuned Florence-2 model, with fewer parameters, can be

deployed on a 16 GB NVIDIA GPU, with an inference cost of approximately $0.0005 per inference and an inference time of around 2.3 seconds per request.

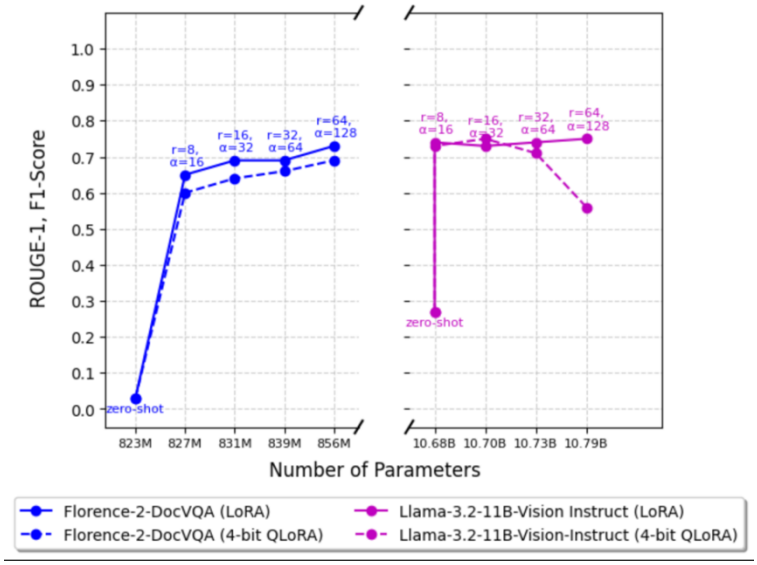

Figure 9. Fine-tuning ROGUE-1, F1-Scores of Florence-2-DocVQA and Llama-3.2-11B-Vision Instruct on the test set after different adaptations.

Table 3.ROUGE similarity scores on a test set of the generated caption on a model adapted to the dataset using LoRA; {P: Precision, R: Recall}. The best results for each metric and each architecture arereflectedin **bold**.

Model	**LoRAConfiguration**	**BLEU Score**	**ROGUE Score Type**	**P**	**R**	**F Score**
Llama-3.2-11B-Vision-Instruct	r = 8, α = 16	0.19	ROUGE-1	0.75	0.76	0.74
			ROUGE-2	0.58	0.60	0.58
			ROUGE-L	0.72	0.73	0.71
	r = 16, α = 32	0.19	ROUGE-1	0.73	0.75	0.73
			ROUGE-2	0.56	0.58	0.56
			ROUGE-L	0.71	0.73	0.71
	r = 32, α = 64	0.19	ROUGE-1	0.75	0.77	0.74
			ROUGE-2	0.57	0.59	0.57
			ROUGE-L	0.72	0.74	0.72
	r = 64, α = 128	**0.21**	ROUGE-1	**0.76**	**0.77**	**0.75**
			ROUGE-2	**0.59**	**0.61**	**0.59**
			ROUGE-L	**0.73**	**0.74**	**0.73**
Florence-2-Doc-VQA	r = 8, α = 16	0.17	ROUGE-1	0.67	0.66	0.65
			ROUGE-2	0.49	0.50	0.48
			ROUGE-L	0.65	0.64	0.63
	r = 16, α = 32	0.20	ROUGE-1	0.72	0.69	0.69
			ROUGE-2	0.55	0.53	0.52
			ROUGE-L	0.70	0.67	0.67
	r = 32, α = 64	0.20	ROUGE-1	0.71	0.70	0.69
			ROUGE-2	0.54	0.54	0.53
			ROUGE-L	0.69	0.68	0.67
	r = 64, α = 128	**0.21**	ROUGE-1	**0.75**	**0.75**	**0.73**
			ROUGE-2	**0.56**	**0.58**	**0.56**
			ROUGE-L	**0.73**	**0.73**	**0.72**

Moreover, as seen in Figure 10, all VLMs in the zero-shot setting struggled with tagging compliance and other logos, with Llama-3.2 performing better on tagging certification logos compared to the other VLMs in this setting. This performance advantage can be attributed to the ground-truth logo tagging data generated using Llama-3.2, which was validated and improved by human annotators. Additionally, the figure shows that adding the adaptation block to Llama-3.2-11B-Vision-Instruct and Florence-2-DocVQA improved the VLMs' performance across all primary logo classes.

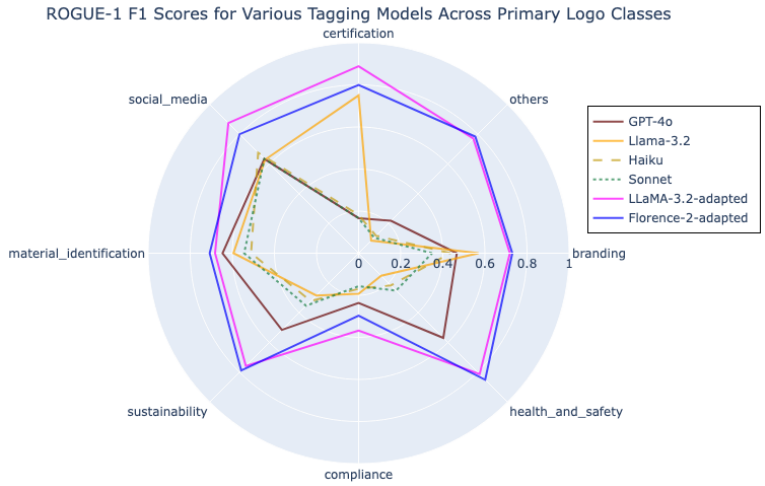

Figure 10. ROGUE-1 F1-Score across primary logo classes for different tagging models. The Llama-3.2 and Florence-2 are adapted models that achieved the best ROGUE-1 F1-scores from the fine-tuning experiments.

3.4. LLM Tagging Performance with OCR Text from Logos

Logos in packaging artwork contain both text and visual information, which makes VLMs a natural choice for logo tagging tasks. However, we also compared logo tagging using LLMs by extracting the OCR text from the logos. Using GPT-4o and Llama-3.2-3B-Instruct LLMs, we evaluated their zero-shot performance, which resulted in BLEU scores of 0.004 and 0.003, respectively. OCR was performed on the logos using PyTesseract. The performance results of these models are summarized in Table 4.

Table 4. “Zero-Shot” performance of LLMs for tag generation using OCR on the test set using text extracted from PyTesseract.

Model	**BLEU Score**	**ROUGE Score Type**	**P**	**R**	**F Score**
Llama-3.2-3B-Vision-Instruct	0.003	ROUGE-1	0.004	0.004	0.004
		ROUGE-2	0.003	0.004	0.003
		ROUGE-L	0.002	0.003	0.002
GPT-4o	0.004	ROUGE-1	0.006	0.005	0.089
		ROUGE-2	0.004	0.005	0.045
		ROUGE-L	0.006	0.008	0.068

3.5. Fine-Tuning Cost

The fine-tuning cost of the Llama-3.2 and Florence-2 models in different LoRA configurations with and without quantization is given in Table 5. All the fine-tuning was performed on an A100 GPU, on the same logo tagging dataset created in this study.

Table 5. Training Time, Cost, and Memory Usage of the Fine-tuned Models for Logo Tagging at Different LoRA Levels {L1: r=8, α=16; L2: r=16, α=32; L3: r=32, α=64; L4: r=64, α=128}.

Model	LoRA	Training Time (sec)	Cost per Hour ($)	Total Cost ($)	Memory Used (MB)
Florence 2 4-bit	L1	1470	3.67	1.499	10764
	L2	1490	3.67	1.519	10868
	L3	1530	3.67	1.560	11040
	L4	1500	3.67	1.529	11290
Florence 2	L1	1120	3.67	1.142	13028
	L2	1130	3.67	1.152	13076
	L3	1130	3.67	1.152	13230
	L4	1130	3.67	1.152	13616
Llama 3.2 4-bit	L1	4368	3.67	4.453	13978
	L2	4387	3.67	4.472	14126
	L3	4385	3.67	4.470	14422
	L4	4392	3.67	4.477	14841
Llama 3.2	L1	3612	3.67	3.682	47464
	L2	3634	3.67	3.705	47750
	L3	3641	3.67	3.712	47922
	L4	3668	3.67	3.739	48476

3.6. Inference Cost

The inference cost of the fine-tuned models in different LoRA configurations with and without quantization is given in Table 6. All the inferencing was conducted on an A100 GPU, on the same logo tagging test dataset created in this study.

Table 6. Training Time, Cost, and Memory Usage of the Fine-tuned Models for Logo Tagging at Different LoRA Levels{L1: r=8, α=16; L2: r=16, α=32; L3: r=32, α=64; L4: r=64, α=128}.

Model	LoRA	Average Inference Time per logo (sec)	Cost per Minute ($)	Average Inference Cost per Logo ($)	Memory Used (MB)
Florence 2 4-bit	L1	0.8	0.042	0.0006	4438
	L2	0.75	0.042	0.0005	4412
	L3	0.75	0.042	0.0005	4500
	L4	0.72	0.042	0.0005	4604
Florence 2	L1	0.75	0.042	0.0005	4404
	L2	0.69	0.042	0.0005	4408
	L3	0.72	0.042	0.0005	4500
	L4	0.72	0.042	0.0005	4604
Llama 3.2 4-bit	L1	0.7	0.042	0.0005	22756
	L2	0.7	0.042	0.0005	22836
	L3	0.7	0.042	0.0005	23020
	L4	0.07	0.042	0.0005	23302
Llama 3.2	L1	0.6	0.042	0.0004	26270
	L2	0.7	0.042	0.0005	26398
	L3	0.7	0.042	0.0005	26630
	L4	0.7	0.042	0.0005	27006

3.6 End-to-End Logo Detection and Tagging Example

Figure 9 shows an end-to-end example output of the proposed cascaded model, where an artwork image is given as input and the logos are detected and tagged.

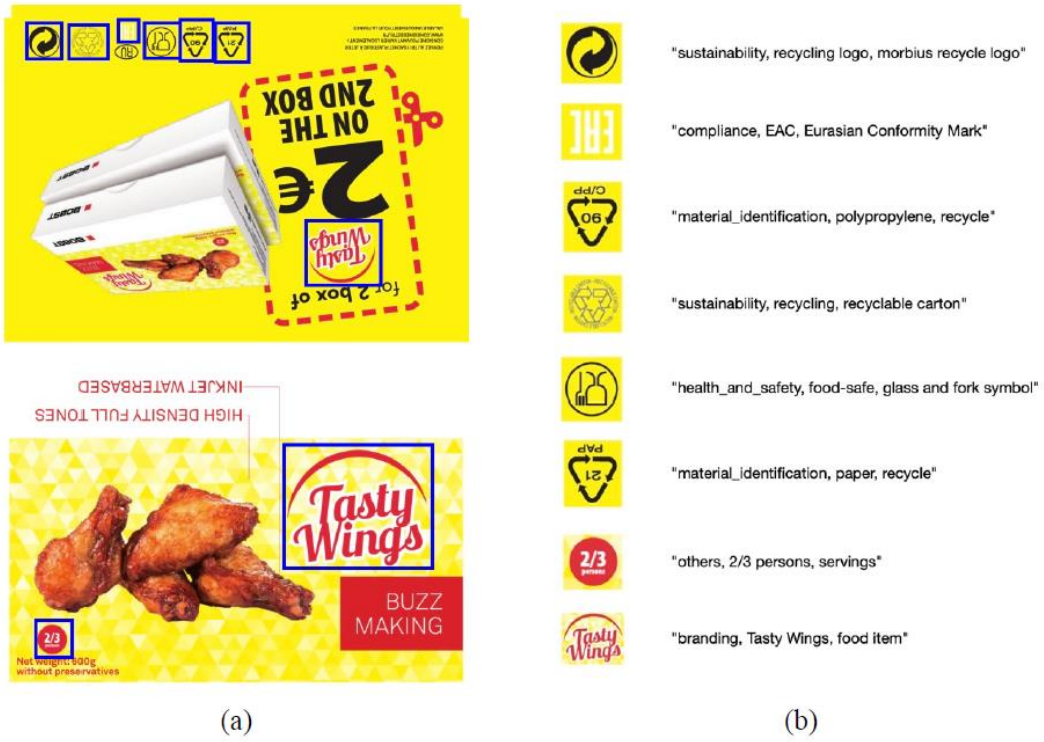

Figure 9. Example output from the model: (a) Cascaded model detecting the different logos present in the artwork file in rectangular bounding boxes; (b) Logos being tagged to further classify into more granular sub-classes

4. CONCLUSION

This study presents a twin-model cascaded framework for logo detection and identification in packaging artwork images. The first phase uses fine-tuned object detection models—YOLOv8, DETR, and Faster-RCNN—to recognize logos and assign primary classifications. The second phase refines tagging with VLMs like Llama-3.2 and Florence-2, improving secondary classification accuracy.

From a cost-efficiency standpoint, smaller models like Florence-2 showed comparable performance to larger models like Llama-3.2 when fine-tuned, offering reduced training time and inference costs without significant accuracy loss.

Validation on custom datasets—Logo Detection and Logo Tagging—demonstrated high annotation quality using human-in-the-loop methods. YOLOv8 achieved the highest mAP of 0.578, outperforming other detection models in terms of inference efficiency. Fine-tuned VLMs, particularly Florence-2, showed notable improvements in tagging accuracy, benefiting from a higher trainable parameter ratio.

This cascaded framework demonstrates the effectiveness of combining lightweight detection and tagging models with fine-tuning for cost-effective solutions. Future work will focus on expanding datasets, incorporating more logo variations, and utilizing humanfeedback reinforcement learning (HFRL) to enhance robustness.

5. LIMITATIONS

Despite its effectiveness, this method is limited by a fixed set of logo categories, restricting adaptability to new logos. The dataset size and diversity also hinder generalization across industries and packaging designs. Performance may suffer with occlusions, distortions, or low-quality images.

Fine-tuning on a smaller dataset can lead to overfitting, while training on a larger dataset is computationally intensive[11]. Additionally, human-in-the-loop annotation introduces biases. Although the system captures subtle logo differences, some variations may be missed, resulting in misclassification.

In conclusion, while the approach shows promise, limitations in dataset diversity, model dependencies, and fine-tuning challenges remain. Future work may focus on expanding datasets, improving robustness, and exploring scalable techniques like HFRL [21].

ACKNOWLEDGEMENTS

The authors would like to thank SGS&COfor providing the opportunity and resources to make this research possible. The authors would also like to thankSGS&CO's Data Science team and Product Managers for enabling this solution.

REFERENCES

[1] H. F. Program, "Guidance for Industry: Food Labeling Guide," FDA, January 2013. [Online]. Available: https://www.fda.gov/regulatory-information/search-fda-guidance-documents/guidance-industry-food-labeling-guide. [Accessed 2025-05-16 2025].

[2] H. Sujuan, L. Jiacheng, M. Weiqing, H. Qiang, Z. Yanna, Z. Yuanjie and J. Shuqiang, "Deep learning for logo detection: A survey," ACM Transactions on Multimedia Computing, Communications and Applications, vol. 20, no. 3, p. 1–23, 2023.

[3] S. Hang, Z. Xiatian and G. Shaogang, "Open logo detection challenge," British Machine Vision Conference, 2018.

[4] Z. Songhui, H. Sujuan and Z. Baisong, "A Decoupled Cross-layer Fusion Network with Bidirectional Guidance for Detecting Small Logos," Proceedings of the 5th ACM International Conference on Multimedia in Asia, p. 1–8, 2023.

[5] M. Daniel Mas, L. Qian, A. Jan and D. Edward, "Scalable logo detection and recognition with minimal labeling," 2018 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR), pp. 152-157, 2018.

[6] H. Steven CH, W. Xiongwei, L. Hantang, W. Yue, W. Huiqiong, X. Hui and W. Qiang, "Logo-net: Large-scale deep logo detection and brand recognition with deep region-based convolutional networks," arXiv preprint, p. arXiv:1511.02462, 2015.

[7] L. Yilin, X. Junke and D. Alireza, "Logo-SSL: Self-supervised Learning with Self-attention for Efficient Logo Detection," in Pacific-Rim Symposium on Image and Video Technology, 2023.

[8] B. Leonid, Logo detection system for automatic image search engines, Google Patents, 2020.

[9] H. Changbo, L. Qun, Z. Zhen, C. Keng-hao and Z. Ruofei, "A multimodal fusion framework for brand recognition from product image and context," 2020 IEEE International Conference on Multimedia & Expo Workshops (ICMEW), pp. 1-4, 2020.

[10] Z. Songhui, H. Sujuan and Z. Baisong, "A Decoupled Cross-layer Fusion Network with Bidirectional Guidance for Detecting Small Logos," Proceedings of the 5th ACM International Conference on Multimedia in Asia, p. 1–8, 2023.
[11] W. Jing, M. Weiqing, H. Sujuan, M. Shengnan, Z. Yuanjie and J. Shuqiang, "LogoDet-3K: A large-scale image dataset for logo detection," ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM), vol. 18, pp. 1--19, 2022.
[12] Y. Sangdoo, H. Dongyoon, O. SeongJoon, C. Sanghyuk, C. Junsuk and Y. Youngjoon, "Cutmix: Regularization strategy to train strong classifiers with localizable features," Proceedings of the IEEE/CVF International Conference on Computer Vision, p. 6023–6032, 2019.
[13] Z. Hongyi, C. Moustapha, D. Yann N and L.-P. David, "mixup: Beyond empirical risk minimization," International Conference on Learning Representations, 2018.
[14] B. Alexey, W. Chien-Yao and L. Hong-Yuan Mark, "Yolov4: Optimal speed and accuracy of object detection," arXiv preprint, p. arXiv:2004.10934, 2020.
[15] OpenAI, "GPT-4 Technical Report," arXiv, p. arXiv.2303.08774, 2024.
[16] M. AI, "Llama 3.2: Revolutionizing edge AI and vision with open customizable models," Meta AI Technical report, 2024.
[17] Anthropic, The Claude 3 Model Family: Opus, Sonnet, Haiku, Anthropic, 2024.
[18] X. Bin, W. Haiping, X. Weijian, D. Xiyang, H. Houdong, L. Yumao, Z. Michael, L. Ce and Y. Lu, "Florence-2: Advancing a unified representation for a variety of vision tasks," Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, p. 4818–4829, 2024.
[19] H. Edward J, S. Yelong, W. Phillip, A.-Z. Zeyuan, L. Yuanzhi, W. Shean, W. Lu, C. Weizhu and others, "LoRA: Low-rank adaptation of large language models," ICLR, vol. 1, no. 2, p. 3, 2022.
[20] D. Tim, P. Artidoro, H. Ari and Z. Luke, "QLoRA: Efficient finetuning of quantized llms," Advances in Neural Information Processing Systems, vol. 36, p. 10088–10115, 2023.
[21] K. Timo, W. Paul, B. Viktor and H. Eyke, "A survey of reinforcement learning from human feedback," arXiv preprint, vol. 10, p. arXiv:2312.14925, 2023.

AUTHORS

Shishir Maurya is a Data Scientist with over 3 years of experience in building advanced AI systems using fine-tuned LLMs and VLMs, and data-driven multi-agent systems. Shishir holds an MS by Research in IT in Building Science from IIIT Hyderabad and a Bachelor of Technology in Aeronautical Engineering. His contributions span multi-agent reinforcement learning, energy informatics, and government-funded AI initiatives, including the Indo-UK RESIDE project (DST India & EPSRC UK). He has authored multiple peer-reviewed international publications.

Anshul Verma is a Data Scientist with over 6 years of experience in developing and deploying AI productions and models. He has worked on multiple AI disciplines, including ML, CV, NLP. He did his Bachelor of Technology in Engineering Physics from Indian Institute of Technology, Madras and holds a Master of Engineering Degree in Electrical and Computer Engineering from University of Toronto.

Yugal Gopal Sharma is a Data Scientist with over 6 years of experience in developing and deploying scalable AI systems. He has built solutions that integrate multiple AI disciplines, including machine learning, CV, NLP, retrieval-augmented generation (RAG), LLMs, and RL. He holds a Bachelor of Technology in Computer Science and Engineering from GGSIPU, Delhi.

Dhanush Dharmaretnam is a Lead AI Solutions Specialist with 10 years of experience in building scalable AI systems across NLP, computer vision, and generative AI. He specializes in multi-agent orchestration, RAG pipelines, and vision-language model fine-tuning. Dhanush holds a Master of Science in Computer Science from the University of Victoria and is a GCP Certified ML Engineer. His work spans enterprise AI applications, with multiple publications and a granted U.S. patent.