Leveraging Computer Vision and Visual LLMs for Cost-Effective and Consistent Street Food Safety Assessment in Kolkata India

Alexey Chernikov[1], Klaus Ackermann[1*], Caitlin Brown[2], Denni Tommasi[3]

[1]SoDa Labs & Department of Econometrics and Business Statistics, Monash University
[2]Department of Economics, Université Laval
[3]Department of Economics, University of Bologna

Abstract

Ensuring street food safety in developing countries is crucial due to the high prevalence of foodborne illnesses. Traditional methods of food safety assessments face challenges such as resource constraints, logistical issues, and subjective biases influenced by surveyors personal lived experiences, particularly when interacting with local communities. For instance, a local food safety inspector may inadvertently overrate the quality of infrastructure due to prior familiarity or past purchases, thereby compromising objective assessment. This subjectivity highlights the necessity for technologies that reduce human biases and enhance the accuracy of survey data across various domains.

This paper proposes a novel approach based on a combination of Computer Vision and a lightweight Visual Large Language Model (VLLM) to automate the detection and analysis of critical food safety infrastructure in street food vendor environments at a field experiment in Kolkata, India. The system utilises a three-stage object extraction pipeline from the video to identify, extract and select unique representations of critical elements such as hand-washing stations, dishwashing areas, garbage bins, and water tanks. These four infrastructure items are crucial for maintaining safe food practices, irrespective of the specific methods employed by the vendors. A VLLM then analyses the extracted representations to assess compliance with food safety standards. Notably, over half of the pipeline can be processed using a user's smartphone, significantly reducing government server workload. By leveraging this decentralised approach, the proposed system decreases the analysis cost by many orders of magnitude compared to alternatives like ChatGPT or Claude 3.5. Additionally, processing data on local government servers provides better privacy and security than cloud platforms, addressing critical ethical considerations. This automated approach significantly improves efficiency, consistency, and scalability, providing a robust solution to enhance public health outcomes in developing regions.

Introduction

Street food safety in developing countries is a significant concern due to its direct impact on public health. The World Health Organization reports over 600 million cases of foodborne illnesses and 420,000 deaths annually from contaminated food, underscoring the urgent need for effective monitoring systems (World Health Organization 2020). Traditional methods, primarily based on human surveys, are time-consuming, prone to biases, and often hindered by the high data collection costs and personnel training (Althubaiti 2016). Street food, with all the associated issues, is nonetheless a crucial source of employment and supply of food for a large proportion of the population (Daniele, Mookerjee, and Tommasi 2021).

This paper explores integrating deep learning and computer vision technologies with visual large language models (VLLMs) to address these challenges and create an automated, scalable solution for assessing street food safety. Specifically, using YOLO for real-time object detection enables the identification of critical features within food vendor environments, such as hand-washing stations, dishwashing areas, garbage bins, and water tanks. Food safety is impossible without available infrastructure, especially in a developing context (Ogwu, Izah, and Ntuli 2024). These detected objects are then analyzed by VLLMs, which provide a detailed assessment of their compliance with food safety standards.

To the best of our knowledge, we are the first to compare large-scale social science survey results with AI models to fill out the survey. This approach enhances the accuracy and speed of safety assessments, removes biases from human surveys, and provides a cost-effective, mobile-friendly solution accessible in resource-constrained settings.

Related Work

Human-based survey The data utilized in this study was collected through a rigorously designed field survey funded by the Food and Agriculture Organization of the United Nations (FAO), a continuation of a long-running field survey (Brown and Tommasi 2024). The survey, conducted by experienced personnel, involved traditional social science methodologies, including structured questionnaires and on-site assessments of food stand infrastructure in Kolkata, India. This survey is a follow-up to a large-scale field experiment conducted two years prior. Such surveys are critical for informing national and international policies to enhance food safety practices, as effective food safety management requires a systemic approach; inspecting every food stand individually is impractical. In this iteration, the sur-

*Corresponding Author: klaus.ackermann@monash.edu

vey maintained the same high standards, including the use of highly trained surveyors, culturally appropriate questions, and expert-led translations. However, a significant addition was made: surveyors were instructed to capture photos and videos of the inspected sites while completing the survey. This innovation allows the survey process to be replayed offline, with the potential for further analysis through AI systems.

ChatGPT 4o The evolution of large language models (LLMs) has driven significant progress in natural language processing (NLP), with OpenAI's GPT series setting new benchmarks in language understanding and generation. GPT-4, an extension of GPT-3, stands out for its improved contextual comprehension, coherence, and ability to generate human-like text (Brown et al. 2020). Trained on diverse datasets, GPT-4o excels in tasks such as text completion, summarization, question answering, and basic reasoning. Beyond text generation, GPT-4o's architecture also supports multimodal tasks, making it effective in analyzing and interpreting textual data in conjunction with visual contexts like image and video descriptions.

Google Paligemma Vision-Language Models (VLMs) have seen significant advancements, beginning with foundational works like CLIP (Radford et al. 2021) and ALIGN (Jia et al. 2021), which established the potential of large-scale image and text embeddings. Subsequent models such as PaLI (Wang et al. 2022), PaLI-X (Wang et al. 2022), and PaLM-E (Wang et al. 2022) further expanded VLM capabilities through generative encoder-decoder architectures, excelling in tasks like image classification, captioning, and VQA. Recent models, including Flamingo (Alayrac et al. 2022) and BLIP-2 (Li et al. 2023), introduced instruction tuning to enhance user interaction and task performance. PaliGemma, the focus of our study, represents the latest evolution in VLMs, integrating the SigLIP-So400m vision encoder with the Gemma-2B language model to deliver competitive performance across a broad range of tasks with a smaller, more efficient architecture.

Florence-2 Florence-2, introduced by (Xiao et al. 2024), stands out as a novel vision foundation model designed to handle a variety of vision and vision-language tasks through a unified, prompt-based representation. This model was trained on the extensive FLD-5B dataset, consisting of 5.4 billion annotations on 126 million images, and demonstrated remarkable capabilities in zero-shot and fine-tuning scenarios. The underlying architecture of Florence-2 integrates a sequence-to-sequence structure, allowing for versatile task performance ranging from object detection to captioning and segmentation.

Current Challenges in Food Safety Assessments Food safety is a global public health issue, particularly pronounced in developing countries where contamination from biological, chemical, and physical hazards can lead to various diseases (World Health Organization 2020). Traditional human survey methods have long been used to gather data on food safety, focusing on consumer awareness, vendor practices, and socioeconomic factors influencing food safety behaviors. However, these methods face several significant limitations, including high costs, logistical challenges, and data inconsistencies. They are resource-intensive, time-consuming, and prone to respondent biases and inaccuracies, which can further exacerbate weak empirical knowledge and institutional fragmentation. Additionally, these surveys may fail to adequately represent remote or underserved populations, leading to skewed data that does not accurately reflect the food safety landscape, ultimately hindering comprehensive food safety interventions (Althubaiti 2016).

Visual Language Models

Visual Language Models (VLMs) combine visual perception with natural language processing to analyze and interpret visual and textual data simultaneously. VLMs can automatically recognize and describe food safety issues by analyzing images of food products and facilities (Ma et al. 2024). This integration of image recognition and natural language processing allows VLMs to provide detailed and accurate assessments quickly, without the biases and fatigue that can affect human surveyors (Radford et al. 2021).

Object Detection: YOLOv10

Object detection has seen significant advancements, with the YOLO (You Only Look Once) series leading due to its balance of efficiency and accuracy. Starting with YOLOv1 (Redmon et al. 2016), subsequent iterations like YOLOv2 and YOLOv3 introduced innovations such as batch normalization and multi-scale predictions (Redmon and Farhadi 2017),(Redmon and Farhadi 2018), while YOLOv4 and YOLOv5 further advanced the field with CSPNet and novel data augmentation techniques (Bochkovskiy, Wang, and Liao 2020; Jocher 2022). The latest, YOLOv10, features NMS-free training and an optimized model design, achieving state-of-the-art performance (Wang et al. 2024). Our work focuses on YOLOv10-N, a specialized variant within the YOLOv10 family, optimized for scenarios requiring minimal latency and efficient parameter use. Despite its compact design, YOLOv10-N maintains competitive accuracy, outperforming other lightweight models on benchmarks like COCO, making it particularly suited for real-time object detection in resource-constrained environments, such as the dynamic conditions of street food markets, where it effectively supports enhanced food safety monitoring.

Feature Extraction With Deep Learning

Deep learning models have revolutionized feature extraction from images, enabling the capture of high-level semantic information. Among these models, Convolutional Neural Networks (CNNs) such as VGGNet, ResNet, and Inception have been widely used for their ability to learn hierarchical features. MobileNet, in particular, has gained attention for its efficiency and effectiveness in mobile and embedded vision applications due to its depthwise separable convolutions, which reduce computational complexity without compromising accuracy (Howard et al. 2017). Other notable feature extractors include DenseNet, which utilizes dense connections between layers to improve feature propagation and

reduce vanishing gradients (Huang et al. 2017). These models have been successfully applied to various tasks, including object detection (Redmon et al. 2016), image segmentation (Ronneberger, Fischer, and Brox 2015), and image classification (Simonyan and Zisserman 2014).

Dataset

Three datasets are used in the framework, each with its specifics and processing procedures.

Field survey Our study utilized a multi-modal approach, integrating traditional survey results with video and image inputs for comprehensive street food safety assessment. To evaluate the quality of the available infrastructure, we devised a series of binary (yes/no) questions, such as, "Is the water storage tank cracked, or does the tank have holes?" see the appendix for the complete questions. These questions served as benchmarks to assess and compare the performance of various Visual Large Language Models (VLLMs). This approach allowed us to gauge the models' accuracy and reliability in identifying infrastructure issues, leveraging visual data and survey responses. Our study analyzed a dataset comprising 7,000 images and 7,510 videos, with responses from 244 out of 328 selected vendors. After applying our inference pipeline, we extracted an additional average of 20 distinct images per video. Given the fixed survey design, which includes field experiments conducted consistently over several years, we maintained the original vendor list for continuity, even though some vendors have reportedly discontinued operations.

Yolo The YOLO dataset utilized the same images generated by the human survey. The training-validation split was conducted by allocating the first 200 vendors with the most images to the training set and the remaining 50 to the validation set. This approach ensured a wide variety of images for training and a diverse range of facilities in the validation set. Initial labelling was performed manually, including the specification of correct bounding box coordinates and class names. The dataset comprised approximately 7000 images for the training set and around 1900 images for the validation set across four classes.

Data augmentation. In our experimental setup, we made specific adjustments to the standard data augmentation parameters used in YOLOv10. To enhance processing efficiency on mobile and edge devices, the image resolution was set to 640 pixels, balancing performance and computational load. We applied Mosaic and MixUp techniques for augmentations, each with a 50% probability and shear augmentation with a 25% probability. These choices were empirically optimized for our specific use case, where shear augmentation was particularly beneficial in simulating diverse angles and perspectives, which is critical given the varied camera angles in our video dataset. All other data augmentation parameters were maintained as per the default YOLOv10 configuration.

Feature extractor and VLLMs training set The primary advantage of a lightweight VLLM is its compact size, which limits its analytical capabilities compared to more robust models like ChatGPT-4 or LLaMA 3. Therefore, the key objective in preparing data for small VLLMs is to maximize the elimination of noise (irrelevant data) to enhance model performance. Additionally, task-specific information needs to be incorporated, such as analyzing the cleanliness of the area surrounding the object. To achieve this, the bounding boxes of detected facilities were expanded by 25%, and all object representations were extracted from the images, and rescaled to ensure one dimension was 640 pixels. This approach removes irrelevant data, leaving only the object and a small portion of its surrounding area for accurate analysis.

Data augmentation. Firstly, images were resized to 224x224 pixels. Next, to enhance the robustness of the feature extractor and VLLM models, data augmentation techniques were employed during training to account for varying lighting conditions encountered during the surveys. These conditions ranged from bright, sunny days to overcast, cloudy weather, affecting the recorded footage's brightness and visibility. To address these variations, the following augmentation methods and parameters were applied: a left-right flip with a 50% probability, an up-down flip with a 50% probability, random brightness adjustments with a ±10% delta, random contrast adjustments within an 80% to 110% range, random hue adjustments with a ±10% delta, and random saturation adjustments within an 80% to 110% range. These augmentations ensured the models could generalize effectively across different environmental conditions.

VLLMs test set The test set used to evaluate the performance of the VLLMs comprised images from 50 vendors that were not included in the training set, as previously described. No validation set was used during the training procedures to prevent data leakage. Similar preprocessing steps were applied to the test set images as with the training set: facilities were detected in each image and cropped with a bounding box enlarged by 25%, then rescaled to ensure the largest dimension was 640 pixels. No data augmentation was applied to the test set images.

Methodology

Brief description The proposed pipeline (Fig. 1) initiates by capturing the video stream from a smartphone or personal camera. Once the recording is complete, the video can be analyzed on a smartphone or PC/laptop. Initially, the system calculates optical flow and selects sharp frames while discarding non-sharp ones, such as during camera relocations. These sharp frames are then batched for recognition using YOLOv10 object detection, identifying and locating objects. Identified objects are batched again for feature extraction, organized by object class (e.g., hand washing stations, garbage bins) to ensure robust analysis from multiple perspectives (angles). MobileNetV3 processes these batches to extract features represented by a feature vector of 64 neurons. Subsequently, these features are clustered into five unique representations per class, ensuring diverse representations by selecting different instances filmed from various angles. Finally, these clustered images are analyzed by VLMs for compliance with food safety standards, using the captured representations to assess and ensure adherence.

This integrated approach significantly minimizes the computational load on government servers, enhances accuracy and security and scales effectively for widespread deployment in resource-constrained environments.

VLLMs

Pipeline The proposed pipeline begins by capturing the video stream from a smartphone or personal camera. Once the recording is finished, the video can be analyzed either on the smartphone or any PC/laptop that supports the ONNX framework (mobile device friendly) The video analysis consists of the following steps:

1. The system calculates optical flow and selects sharp frames, discarding non-sharp frames (e.g., during camera relocation).
2. Each sharp frame is fed into YOLOv10 object detection, where objects are recognized and their location defined.
3. Objects are grouped by class (e.g., hand washing stations, garbage bins) into separate lists for batch feature extraction. This step identifies unique representations of objects for each category — i.e., different instances of the same category (e.g. several garbage bins) and their unique views (from various angles).
4. MobileNetV3 extracts their features in batches for each list from the previous step. The feature vector for each object consists of 64 neurons.
5. All the extracted features are then clustered into 5 clusters to obtain five unique representations for each category by calculating the closest feature set to each cluster center. Different instances for each category filmed from various angles are selected.
6. The resulting set of images (the number of categories in the video multiplied by 5) is sent to the VLLM.
7. By category, VLLM analyses the given set of images.

The survey data consists of yes/no/unknown responses, which lack the detailed information necessary for training VLLMs. Therefore, to generate explanations and reasoning for the answers, we utilized ChatGPT4o. This service offers accurate answers accompanied by concise reasoning, adhering to the provided answer template.

ChatGPT-4 Analysis **Textual Data Processing**: The textual data from the questionnaires was preprocessed and fed into ChatGPT-4. The model analysed the responses and provided insights on street food safety practices. **Image and Video Analysis**: Selected images and video frames were described and analyzed using ChatGPT-4. The model generated textual descriptions and answered specific survey questions about each visual input. **Comparison of Insights**: The responses generated by ChatGPT-4 were then compared with the human survey responses to evaluate the model's accuracy and reliability.

Google Paligemma The core of our approach involves fine-tuning PaliGemma, a robust Vision-Language Model, to specialize in answering survey questions. It was trained on the answers previously extracted by ChatGPT4o. PaliGemma's architecture consists of a SigLIP image encoder and a Gemma-2B language model, providing a solid foundation for our fine-tuning tasks.

Image Encoder: SigLIP-So400m The SigLIP-So400m model is a vision encoder optimized for shape recognition and contrastive learning. It processes images into a sequence of 400 million tokens, which are then fed into the language model. This encoder has shown state-of-the-art performance in various vision tasks, making it an ideal choice for our VLM.

Language Model: Gemma-2B The Gemma-2B is a decoder-only language model with 2 billion parameters, built using advanced autoregressive techniques. It generates coherent and contextually relevant text based on the input image and text prompts provided by the encoder.

Integration of Image and Text Tokens The image tokens generated by the SigLIP encoder are projected into the same dimension as the Gemma-2B tokens through a linear projection layer. These tokens are concatenated with text tokens from the SentencePiece tokenizer, forming a unified sequence for the language model to process.

Florence-2 Florence-2 employs a unified sequence-to-sequence learning paradigm to handle various vision tasks. The model's architecture integrates an advanced vision encoder, DaViT (Ding et al. 2022), with a multi-modality transformer-based encoder-decoder. The vision encoder processes input images into visual token embeddings, combined with text embeddings derived from task-specific prompts. The multi-modality encoder-decoder processes this combined input to generate text-based outputs. As Paligemma it was also trained on the answers provided by ChatGPT4o.

Surveyor (Inference) Pipeline

YOLOv10

Fitness The fitness function in YOLO models, a composite loss function, is vital to optimize object detection accuracy. It integrates localization, confidence, and class probability losses to measure the discrepancy between predicted and ground truth bounding boxes and class probabilities. Minimizing this function improves the model's object detection and classification performance. The key evaluation metrics are mAP50 and mAP50-90. mAP50 measures mean Average Precision (mAP) at a 50% IoU threshold, while mAP50-90 averages mAP across IoU thresholds from 50% to 90%, offering a broader evaluation. YOLO 10 uses fitness weights [0,0,0.1,0.9] for Precision, Recall, mAP50, and mAP50-90, respectively, emphasizing mAP50-90 in performance evaluation.

Loss Function The loss function in YOLOv10 is designed to optimize the balance between localization accuracy and classification performance. It comprises three main components: the bounding box regression loss, the objectness score loss, and the classification loss.

Bounding Box Regression Loss: This component minimizes the error in predicting the coordinates of the bounding boxes. It uses a smooth L1 loss, which is less sensitive

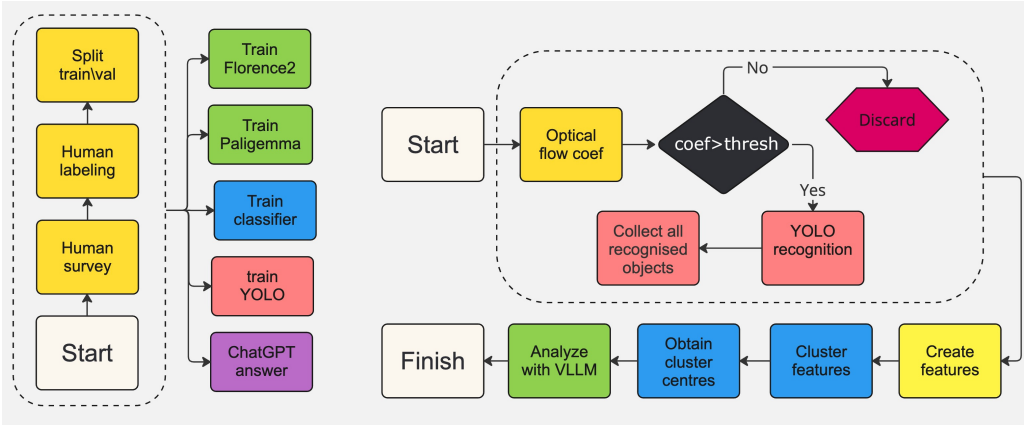

Figure 1: Training flowchart of the proposed framework on the left, inference pipeline - on the right.

to outliers than the standard L2 loss. The regression loss is given by:

$$\mathcal{L}_{\text{reg}} = \sum_{i=1}^{S^2} \sum_{j=1}^{B} 1_{ij}^{\text{obj}} \big((t_x - \hat{t}_x)^2 + (t_y - \hat{t}_y)^2 + (t_w - \hat{t}_w)^2 + (t_h - \hat{t}_h)^2 \big) \quad (1)$$

where t_x, t_y, t_w, t_h are the predicted coordinates and dimensions of the bounding box, and $\hat{t}_x, \hat{t}_y, \hat{t}_w, \hat{t}_h$ are the ground truth values.

Objectness Score Loss: This component penalizes the model for incorrect predictions about the presence of an object in a grid cell. It uses binary cross-entropy loss to achieve this:

$$\mathcal{L}_{\text{obj}} = \sum_{i=1}^{S^2} \sum_{j=1}^{B} 1_{ij}^{\text{obj}} \left(\log \hat{p}_{\text{obj}} + (1 - \hat{p}_{\text{obj}}) \log(1 - \hat{p}_{\text{obj}}) \right)$$

where $\hat{p}_{\text{obj}}$ is the predicted probability of an object being present in the bounding box.

Classification Loss: This component addresses the error in predicting the class of the detected object. It also uses a cross-entropy loss:

$$\mathcal{L}_{\text{cls}} = \sum_{i=1}^{S^2} 1_{i}^{\text{obj}} \sum_{c \in \text{classes}} \left(p_c \log \hat{p}_c + (1 - p_c) \log(1 - \hat{p}_c) \right)$$

where p_c is the true class probability and $\hat{p}_c$ is the predicted class probability.

The overall loss function is a weighted sum of these three components:

$$\mathcal{L} = \lambda_{\text{reg}} \mathcal{L}_{\text{reg}} + \lambda_{\text{obj}} \mathcal{L}_{\text{obj}} + \lambda_{\text{cls}} \mathcal{L}_{\text{cls}}$$

where $\lambda_{\text{reg}}, \lambda_{\text{obj}}, \lambda_{\text{cls}}$ are the weights for each component.

Facility Representation Extraction From Video Feed

To extract distinctive representations of the facilities captured in the video feed, we adopted an approach inspired by the method introduced by (Chernikov et al. 2022). The method has been tailored to suit our specific requirements for analyzing the video content.

Feature extraction: MobileNet. MobileNetV3 introduces an optimized neural network architecture for mobile and resource-constrained environments, leveraging advanced techniques like neural architecture search (NAS) and NetAdapt to achieve an optimal balance between accuracy and efficiency. The architecture employs a combination of inverted residual blocks with linear bottlenecks, enhanced by squeeze-and-excitation modules and an innovative h-swish activation function, which is more efficient for mobile CPUs. The network is designed with two versions, MobileNetV3-Large and MobileNetV3-Small, targeting different resource levels. MobileNetV3 demonstrates superior performance in tasks such as image classification, object detection, and semantic segmentation, outperforming previous models like MobileNetV2 and MnasNet, while significantly reducing latency and computational costs, making it highly suitable for mobile applications (Howard et al. 2019).

Feature extraction is performed using the penultimate dense layer of the model, which consists of 64 neurons and follows the global average pooling layer. Consequently, each image produces two outputs during the inference stage: a class probability vector and a feature vector.

Feature Clustering with KMeans After recognizing objects from the video and extracting features, we reduce the number of images sent to the VLLM by clustering them, targeting 5 per class. K-Means clustering, chosen for its simplicity and effectiveness, partitions data into k clusters by minimizing within-cluster variance. It starts with initializing k centroids, assigning data points to the nearest centroid, and iterating until convergence. We use K-Means++ initialization to improve clustering by better centroid placement (Arthur and Vassilvitskii 2007).

Experimental Study

Hardware setup Given the lightweight nature of the selected Vision-Language Models (VLMs), the computational hardware requirements for our experiments were modest. The training was conducted on an AMD Threadripper 3955WX with 128 GB of RAM and dual NVidia GeForce RTX 4090 GPUs. For inference, we utilized a system

equipped with an Intel 12600K processor, 64 GB of RAM, and a single NVidia GeForce RTX 3090 GPU.

Google PaliGemma Fine-Tuning We fine-tuned PaliGemma on our survey dataset to enhance its ability to generate accurate answers from visual and textual inputs. Key hyperparameters included a batch size of 8, 30,000 training steps, and a maximum of 128 new tokens, using a 224-pixel checkpoint. Fine-tuning was done in JAX using the bfloat16 (bf16) data type, which balances range and precision. Both Vision (img/) and attention layers (llm/layers/attn/) were set as "trainable."

Florence-2 For a fair comparison, the fine-tuning process for Florence-2 was closely aligned with the approach used for Paligemma. However, there were a few key differences. We utilized the same training and validation datasets, maintaining an image resolution of 224x224 pixels. Unlike Paligemma, which employed the SGD optimizer, we opted for AdamW, ensuring more stable and faster convergence. Additionally, rather than using JAX, as in Paligemma's case, our training was conducted using PyTorch. Another distinction was the use of the float16 data type during training, which contributed to efficient computation. We selected a linear learning rate scheduler without warmup and trained the model over 30 epochs. Due to Florence-2's slightly larger complexity compared to Paligemma, we were limited to batches of up to 6 samples during training. Finally, we set the max_new_tokens parameter to 128 tokens.

YOLOv10

Fitness In our framework, object detection constitutes the most critical step of the pipeline. Incorrect detection, such as misclassifying an object, leads to the VLLM analyzing an erroneous object, potentially resulting in harmful outcomes. Given that we work with videos and have an ample supply of images, the primary objective in object detection is to ensure the correct classification of images, emphasizing Precision over Recall. Consequently, we modified the fitness function of the YOLO training pipeline from [0,0,0.1,0.9] to [0.5,0,0.1,0.5]. This adjustment prioritizes Precision, ensuring that all detected objects are correctly classified and suitable for subsequent VLLM analysis, while a reduction in Recall is acceptable in our context.

Train hyperparameters We trained our YOLOv10 model on the COCO dataset, optimizing the loss function described in the Methodology section. As mentioned, our focus in this study is on maximizing precision over recall. To achieve the highest possible precision, we significantly increased the weight of the classification loss in the total loss function, raising it from 0.5 to 6. Concurrently, we reduced the weight of the box loss component from 7.5 to 1. These adjustments were made to fine-tune the model's performance towards our precision-centric objectives. We employed the Stochastic Gradient Descent (SGD) optimizer for optimisation, which is known for its superior generalization capabilities despite its relatively slower convergence. The SGD parameters were kept at their standard values, with a learning rate 1E-2 and momentum set to 0.97. The model underwent

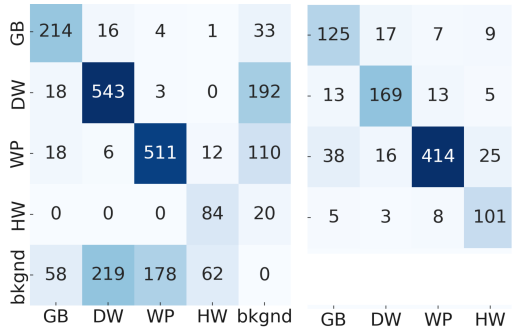

Figure 2: (A) Performance metrics of YOLOv10 object detection; (B) Resulting performance metrics of the feature extractor/classifier.

training for 500 epochs to ensure adequate learning and performance stability.

Feature extractor The model was trained over 5000 epochs with a batch size of 1024 samples, utilizing the SGD optimizer. To enhance generalization, Label Smoothing with a coefficient of 0.1 was applied. A dropout layer with a coefficient of 75% was incorporated immediately after the Global Average Pooling layer, before the feature extraction layer, to combat overfitting. Additionally, a Reduce on Plateau Learning Rate scheduler with a multiplier of 0.66 and a patience of 150 epochs was employed to improve generalization further. Since the dataset was pre-balanced, standard categorical cross-entropy was used as the loss function.

Results

Inference Pipeline

A significant challenge in our work stemmed from the unregulated and highly variable appearance of facilities, which differs considerably from vendor to vendor. This variation becomes even more pronounced when a single vendor operates multiple entities of the same facility type that vary in appearance, such as garbage bins, ranging from plastic bags to repurposed water tanks. Despite this variability, modern computer vision architectures have performed well in addressing this task.

Object detection Fig. 2A demonstrates that the model achieves high precision across all classes, effectively minimizing the misclassification of items as other categories. This high precision, while resulting in some underclassification of items as background, is a strategic trade-off that aligns with our primary goal: ensuring that when items are detected, they are not misclassified. This approach significantly reduces false positives, a critical factor in further VLLM analysis.

Features extractor As it seen in Fig. 2B, MobileNetV3 demonstrated strong performance despite the variability in representations. Some misclassification occurred, particularly between water tanks and hand washing stations, which is understandable given that hand washing stations often use similar water tanks on top of them. However, the model's

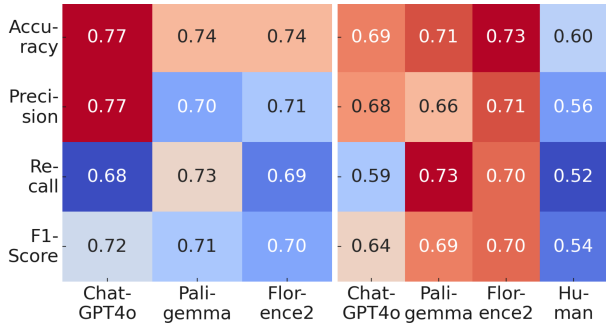

Figure 3: (A) Performance metrics of fine-tuned VLLMs on human survey data; (B) Metrics for disagreements only.

high accuracy is not critical in this context, as its primary role is to differentiate the most prominent representations of the facilities.

Extracted Representation Analysis With VLLMs

VLLMs were evaluated on a test set comprising 258 images across four facility types, with a total of 864 questions administered.

Full test set results with an expert supervision Fig. 3 demonstrates the final results obtained for 3 Vision LLM models on a LLM test set: ChatGPT, Paligemma, Florence-2. The results were obtained by comparing human survey data with predictions of the ChatGPT and fine-tuned VLLMs. Cases where there were disagreements between the human survey and ChatGPT were double-checked and corrected by a field expert. The expert did not control cases with an agreement between the survey team and ChatGPT4o. Therefore, the ground truth in the experiment was a mixture of the survey team's opinion (in cases with full agreement) and an expert opinion (in cases with disagreements). ChatGPT4o slightly outperforms the other VLLM models, with Paligemma and Florence2 closely following in overall performance metrics.

Disagreements A total number of cases with disagreement between the survey team and ChatGPT4o was 250 out of 780. In this experiment, the ground truth was solely an expert opinion. We removed cases with a full agreement to investigate the performance of the survey team, ChatGPT40 and our fine-tuned models. Fig. 3A demonstrates that the human survey team had the lowest scores across all evaluation metrics. The figure demonstrates that VLLM models, particularly Florence2, generally outperform the human survey team in terms of accuracy, precision, recall, and F1 score, highlighting the effectiveness of automated approaches in food safety assessment.

Inference time The inference speeds were measured using the hardware setup detailed in the Methodology section.

The maximum batch size that can be processed on a single NVIDIA GeForce RTX 3090 varies by model. In our evaluation, ChatGPT took 30 minutes for inference, while PaliGemma demonstrated faster performance with larger batch sizes, achieving 154.24 seconds at a batch size of 16. In contrast, Florence2 was most efficient at smaller batch sizes, with a time of 178.42 seconds at a batch size of 6.

Figure 4: (A) Handwashing facility: 10th May vs. 24th July 2024. AI correctly flags a missing lid and soap (contamination risk), while human reports both present, likely due to survey fatigue. (B) Dishwashing facilities: AI misidentifies the washing location; the surveyor's assessment is accurate.

Conclusion

We have developed a robust framework for automated street food safety surveys, which effectively mitigates biases, reduces reliance on domain expertise, and can be implemented by surveyors with minimal training. By primarily utilizing edge/mobile devices, this framework minimizes the need for robust infrastructure, leading to significant cost savings. While errors are inevitable in manual and automated approaches—illustrated by 4A, where a human inspector made an error, and 4B, where the AI model misinterpreted height—our system's design allows continuous optimization and improvement. Although we retained the AI model's error in our analysis to highlight potential pitfalls, future optimization will address such issues. The framework's scalable design enables broad geographical application and enhanced analytical depth, making it adaptable for future expansions in the list of analyzed facilities. This tool is intended to empower the same survey team to assess an expanded set of food vendors in subsequent surveys, offering local governments a comprehensive understanding of infrastructure gaps in various city areas. As a future direction, we aim to scale this procedure to 100 cities across India, evaluating the requirements for expansion. Ultimately, this framework's inference allows street food surveyors to conduct real-time, automated assessments via smartphones, potentially reducing the prevalence of foodborne diseases significantly.

Acknowledgements

We thank FAO Italy (Cornelia Boesch, William Hallman [Rutgers University], Jeffrey Lejeune, Markus Lipp, Keya Mukherjee, Diletta Topazio, Kang Zhou) and FAO India (Vinay Singh, Takayuki Hagiwara) for their support. We also appreciate our local NGO READS, data collector NYAS, and funding from the FAO Innovator Fund. Additionally, we thank Google for providing academic cloud research credits.

References

Alayrac, J.-B.; Donahue, J.; Luc, P.; Miech, A.; Barr, I.; Hasson, Y.; Lenc, K.; Mensch, A.; Millican, K.; Reynolds, M.; Ring, R.; Rutherford, E.; Cabi, S.; Han, T.; Gong, Z.; Simonyan, K.; et al. 2022. Flamingo: a visual language model for few-shot learning. *arXiv preprint arXiv:2204.14198*.

Althubaiti, A. 2016. Information bias in health research: Definition, pitfalls, and adjustment methods. *Journal of Multidisciplinary Healthcare*, 9: 211–217.

Arthur, D.; and Vassilvitskii, S. 2007. K-Means++: The Advantages of Careful Seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, 1027–1035. Society for Industrial and Applied Mathematics.

Bochkovskiy, A.; Wang, C.-Y.; and Liao, H.-Y. M. 2020. YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv preprint arXiv:2004.10934*.

Brown, C.; and Tommasi, D. 2024. Quality Upgrading in the Street Food Market: Are Better Infrastructure and Training Sufficient? Working Paper.

Brown, T. B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. 2020. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33: 1877–1901.

Chernikov, A.; Tan, C. W.; Montero-Manso, P.; and Bergmeir, C. 2022. FRANS: Automatic Feature Extraction for Time Series Forecasting.

Daniele, G.; Mookerjee, S.; and Tommasi, D. 2021. Informational Shocks and Street-Food Safety: A Field Study in Urban India. *The Review of Economics and Statistics*, 103(3): 563–579.

Ding, M.; Xiao, B.; Codella, N.; Luo, P.; Wang, J.; and Yuan, L. 2022. Davit: Dual attention vision transformers. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXIV*, 74–92. Springer.

Howard, A.; Sandler, M.; Chu, G.; Chen, L.-C.; Chen, B.; Tan, M.; Wang, W.; Zhu, Y.; Pang, R.; Vasudevan, V.; et al. 2019. MobileNetV3: An optimized neural network architecture for mobile and resource-constrained environments. *arXiv preprint arXiv:1905.02244*.

Howard, A. G.; et al. 2017. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *arXiv preprint arXiv:1704.04861*.

Huang, G.; Liu, Z.; Van Der Maaten, L.; and Weinberger, K. Q. 2017. Densely Connected Convolutional Networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4700–4708.

Jia, C.; Yang, Y.; Xia, Y.; Chen, Y.-T.; Parekh, Z.; Pham, H.; Le, Q. V.; Sung, Y.-H.; Li, Z.; and Duerig, T. 2021. Scaling up visual and vision-language representation learning with noisy text supervision. In *International Conference on Machine Learning*, 4904–4914. PMLR.

Jocher, G. 2022. YOLOv5. Retrieved from https://github.com/ultralytics/yolov5, Accessed: 2024-06-01.

Li, J.; Li, D.; Chen, C.; Zhang, P.; Zhang, K.; Wang, L.; Yuan, L.; et al. 2023. BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models. *arXiv preprint arXiv:2301.12597*.

Ma, P.; Tsai, S.; He, Y.; Jia, X.; Zhen, D.; Yu, N.; Wang, Q.; Ahuja, J. K.; and Wei, C.-I. 2024. Large language models in food science: Innovations, applications, and future. *Trends in Food Science Technology*, 148: 104488.

Ogwu, M. C.; Izah, S. C.; and Ntuli, N. R. 2024. Food Safety and Quality in the Global South.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. 2021. Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020*.

Redmon, J.; Divvala, S.; Girshick, R.; and Farhadi, A. 2016. You Only Look Once: Unified, Real-Time Object Detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Redmon, J.; and Farhadi, A. 2017. YOLO9000: Better, Faster, Stronger. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Redmon, J.; and Farhadi, A. 2018. YOLOv3: An Incremental Improvement. *arXiv preprint arXiv:1804.02767*.

Ronneberger, O.; Fischer, P.; and Brox, T. 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, 234–241.

Simonyan, K.; and Zisserman, A. 2014. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv preprint arXiv:1409.1556*.

Wang, A.; Chen, H.; Liu, L.; Chen, K.; Lin, Z.; Han, J.; and Ding, G. 2024. YOLOv10: Real-Time End-to-End Object Detection. *arXiv preprint arXiv:2405.14458*.

Wang, P.; Hu, H.; Zhang, Z.; Hu, X.; et al. 2022. PaLI: A Large-Scale Multilingual Vision-Language Model. *arXiv preprint arXiv:2206.09209*.

World Health Organization. 2020. Food safety. *WHO*.

Xiao, B.; Wu, H.; Xu, W.; Dai, X.; Hu, H.; Lu, Y.; Zeng, M.; Liu, C.; and Yuan, L. 2024. Florence-2: Advancing a Unified Representation for a Variety of Vision Tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE.