How Does India Cook *Biryani*?

C V Rishi[†]
IIIT Hyderabad
India
se23ucse044@mahindrauniversity.edu.in

Farzana S[†]
IIIT Hyderabad
India
farzana.s@research.iiit.ac.in

Shubham Goel[†]
IIIT Hyderabad
India
shubham.goel@students.iiit.ac.in

Aditya Arun
IIIT Hyderabad
India
adityaarun1@gmail.com

C V Jawahar
IIIT Hyderabad
India
jawahar@iiit.ac.in

Abstract

Biryani, one of India's most celebrated dishes, exhibits remarkable regional diversity in its preparation, ingredients, and presentation. With the growing availability of online cooking videos, there is unprecedented potential to study such culinary variations using computational tools systematically. However, existing video understanding methods fail to capture the fine-grained, multimodal, and culturally grounded differences in procedural cooking videos. This work presents the first large-scale, curated dataset of *biryani* preparation videos, comprising 120 high-quality YouTube recordings across 12 distinct regional styles. We propose a multi-stage framework leveraging recent advances in vision–language models (VLMs) to segment videos into fine-grained procedural units and align them with audio transcripts and canonical recipe text. Building on these aligned representations, we introduce a video comparison pipeline that automatically identifies and explains procedural differences between regional variants. We construct a comprehensive question–answer (QA) benchmark spanning multiple reasoning levels to evaluate procedural understanding in VLMs. Our approach employs multiple VLMs in complementary roles, incorporates human-in-the-loop verification for high-precision tasks, and benchmarks several state-of-the-art models under zero-shot and fine-tuned settings. The resulting dataset, comparison methodology, and QA benchmark provide a new testbed for evaluating VLMs on structured, multimodal reasoning tasks and open new directions for computational analysis of cultural heritage through cooking videos. We release all data, code, and the project website at https://farzanashaju.github.io/how-does-india-cook-biryani/.

CCS Concepts

• **Computing methodologies** → **Computer vision**; **Scene understanding**; **Activity recognition and understanding**; *Video segmentation.*

Keywords

Video Understanding, Vision Language Models

ACM Reference Format:
C V Rishi, Farzana S, Shubham Goel, Aditya Arun, and C V Jawahar. 2025. How Does India Cook *Biryani*?. In *Indian Conference on Computer Vision, Graphics, and Image Processing (ICVGIP 2025), December 17–20, 2025, Mandi, India.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3774521.3774596

1 Introduction

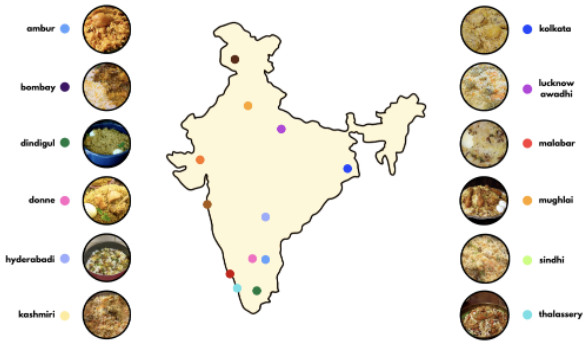


Figure 1: Map of India showing 12 regional biryani types - Ambur, Bombay, Dindigul, Donne, Hyderabadi, Kashmiri, Kolkata, Awadhi, Malabar, Mughlai, Sindhi, and Thalassery. Representative images illustrate differences in preparation, ingredients, and presentation, with all videos sourced from YouTube to capture authentic regional cooking practices.

Biryani is more than a culinary dish; it is a cultural symbol that embodies the diversity and richness of Indian gastronomy. While its name is shared across the country, its preparation varies widely across regions, shaped by local traditions, availability of ingredients, and individual cooking styles. These differences manifest in flavour and the sequence of preparation steps, the choice of utensils, and the presentation style. With the proliferation of online platforms such as YouTube, this diversity is now documented at scale through cooking videos, providing an invaluable record of culinary practices. However, despite abundant content, the computational tools required to systematically capture and compare fine-grained procedural variations in such videos remain underdeveloped.

[†]Equal Contribution

ICVGIP 2025, Mandi, India

ACM ISBN 979-8-4007-1930-1/2025/12
https://doi.org/10.1145/3774521.3774596

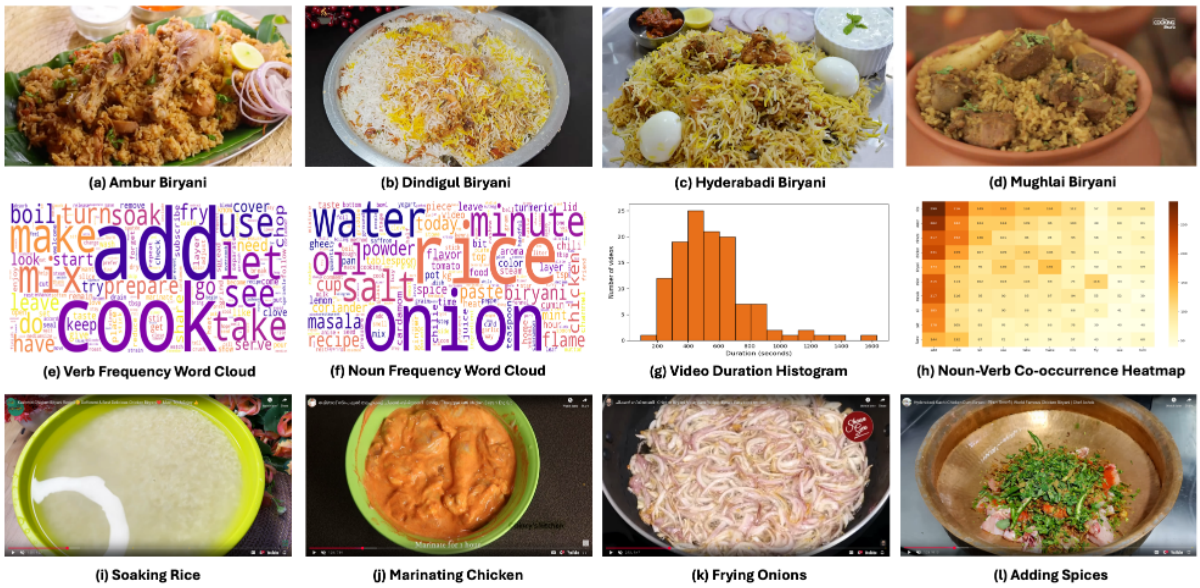


Figure 2: Overview of the Biryani Dataset. Panels (a–d) show representative frames from four of the twelve biryani categories - Ambur, Dindigul, Hyderabadi, and Mughlai - capturing regional diversity in presentation, colour palette, and plating. Panels (e) and (f) present verb and noun frequency word clouds derived from ASR-transcribed and translated speech, revealing common procedural actions and key ingredients. Panel (g) shows the distribution of video durations, with most videos between 5-12 minutes, while panel (h) visualises a noun–verb co-occurrence heatmap, highlighting frequent action–ingredient pairings central to biryani preparation. Panels (i–l) depict canonical procedural steps identified via GPT-4-generated template recipes.

Cooking videos present a unique challenge for computer vision due to their multimodal nature, temporal complexity, and diversity in visual presentation [8, 17, 21, 28, 43, 46]. The same high-level dish can be prepared using markedly different sequences of actions, ingredient combinations, and stylistic elements, often accompanied by narration in different languages or dialects. Indian cooking is known for its multi-step processes and intricate use of spices, and these details are central to understanding the cultural and procedural identity of a recipe [1, 4, 33]. Conventional video understanding approaches have primarily focused on coarse-grained action recognition or highlight detection, which are insufficient for modelling such nuanced, structured tasks [10, 15, 21, 26].

Over the past two decades, video analysis has evolved from handcrafted feature-based methods [3, 23], such as Hidden Markov Models [19, 24, 41] and Support Vector Machines [6, 9, 22], to deep learning models capable of capturing richer visual patterns from large datasets [25, 27, 32]. More recently, large vision–language models (VLMs) have emerged as a powerful paradigm, jointly reasoning over visual and textual information to produce semantically meaningful outputs [20, 34, 47]. These models have demonstrated strong generalisation capabilities in diverse domains, yet their application to structured procedural understanding remains relatively unexplored, particularly in culturally rich contexts. In the context of cooking, and *biryani* in particular, VLMs can move beyond recognising individual actions toward modelling the full procedural flow, aligning it with textual recipes, and enabling fine-grained comparisons between variations.

The contributions of this work are as follows:

- We introduce the first curated dataset of Indian *biryani* preparation videos, annotated with fine-grained temporal segmentation and multimodal labels.
- We design a robust VLM-based pipeline for procedural video segmentation, multimodal alignment, and question-answer generation.
- We propose a novel video comparison framework for analysing subtle procedural differences across regional *biryani* variants.
- We provide quantitative benchmarks and qualitative analyses of the performance of current VLMs on culturally rich procedural video understanding tasks.

The remainder of the paper is organised as follows. Section 2 describes the dataset curation process, Section 3 details the video segmentation framework, Section 4 presents the multimodal alignment methodology, Section 5 outlines the QA dataset generation and benchmarking experiments, Section 6 introduces the video comparison framework, and Section 7 discusses potential applications and concludes.

2 Biryani Dataset

We want to study how different videos curated for the same purpose (in this case cooking *biryani*) differs or compares. We start with creating a dataset of publicly available *biryani* cooking videos. We curate a dataset of 120 cooking videos focused on *biryani* preparation, sourced from YouTube. The dataset spans 12 distinct types of *biryani* (Ambur, Bombay, Dindigul, Donne, Hyderabadi, Kashmiri,

Kolkata, Awadhi, Malabar, Mughlai, Sindhi, and Thalassery). For each category, we collect 10 distinct videos per category, as shown with representative frames in Figure 2 (a-d), illustrating the diversity in presentation, colour palette, and plating traditions across regions.

Videos were chosen for their culinary popularity and the availability of high-quality recordings. To maximise utility for the downstream tasks, we prioritized videos featuring clear audio, spoken narration of cooking steps, complete visual coverage of the preparation process, and a range of durations. Given the pan-Indian diversity of the selected *biryani* types, the dataset exhibits substantial variation in language, cooking techniques, narration styles, and cinematographic choices such as camera angles and editing styles.

We first extract audio from each video and perform automatic speech recognition (ASR) using WhisperX [5] and Whisper-Large [29]. All transcripts are translated into English (using GPT-4 [2]) to standardise linguistic representation across the dataset. We then use part-of-speech tagging with spaCy [12] to extract nouns, verbs, and adjectives from the transcripts, producing frequency-based visualisations such as word clouds. Figures 2 (e, f) show examples of these visualisations, where high-frequency verbs (e.g., "add", "cook") and nouns (e.g., "rice", "onion") capture the procedural and ingredient focus of *biryani* preparation. Additional analyses, such as the duration histogram in Figure 2 (g), reveal that most videos fall within a 5–12 minute range, while the noun–verb co-occurrence heatmap in Figure 2 (h) highlights common action–object pairings that define core cooking steps.

To enable fine-grained analysis (such as step-level captioning or instruction grounding), we segment each cooking video into meaningful procedural units. We generate canonical template recipes for each *biryani* type using GPT-4 [2], which provided structured reference sequences of cooking steps. These generated templates served as a standardized framework for identifying procedural steps across diverse video formats, rather than acting as an authentic recipe ground-truth. Manual verification ensured the consistency and usability of this framework for temporal segmentation. An additional "Miscellaneous/Intro/Outro" class is used in each template to account for non-cooking content commonly present in YouTube videos, such as greetings, personal anecdotes, promotional messages, or outros, ensuring that such segments are meaningfully grouped and excluded from step-level misalignment. Figure 2 (i–l) depicts canonical procedural frames extracted from videos, including soaking rice, marinating chicken, frying onions, and adding spices—steps that recur across multiple *biryani* variants despite regional differences.

3 Video Segmentation

We use InternVL-14B [48], a state-of-the-art Vision-Language Model (VLM), to process each segment. As shown in Fig. 3, the model is prompted to extract three key categories of information: (a) Ingredients, (b) Utensils (Objects), and (c) Actions (verbs). Significantly, the model relies solely on visual content (sampled video frames) and does not access audio or transcripts, ensuring that annotations are grounded purely in visual evidence.[1] Since cooking actions often span more than one 10-second interval, the same canonicalised action label can appear in consecutive segments. To improve temporal coherence, we merge timestamps for such repeated actions within a video into a single continuous span, while ensuring unrelated actions in adjacent segments remain separate. This reduces unnecessary fragmentation and yields longer, coherent action-level sequences without merging distinct activities. Direct application of InternVL-14B across thousands of segments yields a detailed mapping of ingredients, utensils, and actions over time. However, action descriptions often vary lexically despite being semantically identical (e.g., "stirring rice" vs. "stirring rice and water with a wooden spoon"). To address this, each action phrase is embedded using the all-MiniLM-L6-v2 SentenceTransformer model [30] and clustered via agglomerative clustering with average linkage and a cosine distance threshold of 0.3, merging clusters until no pair falls below this threshold. A single representative phrase from each cluster serves as the canonical action label, improving label consistency and enabling robust querying, statistical analysis, and downstream tasks such as recipe step generation and video retrieval.

Although InternVL-14B produced high-quality visual annotations, we introduced an automated verification step using Gemini-2.5-flash-lite [35] to ensure each labelled action was visibly present in its corresponding segment. This lightweight VLM was queried with deterministic yes/no prompts over sampled video frames, enabling reliable validation for downstream tasks such as step-wise recipe alignment and skill-specific retrieval. [2]. We verified 14,470 video–action segments across all *biryani* types, with 11,295 (78.05%) labelled as correct and 3,175 (21.95%) as incorrect, thereby increasing confidence in the dataset's action labels.

3.1 Results

The initial action detection stage produced a highly granular label space, with 10,481 unique action classes. After applying the action clustering process, this number was reduced to 2,187 canonicalised action classes, greatly improving consistency in labelling.

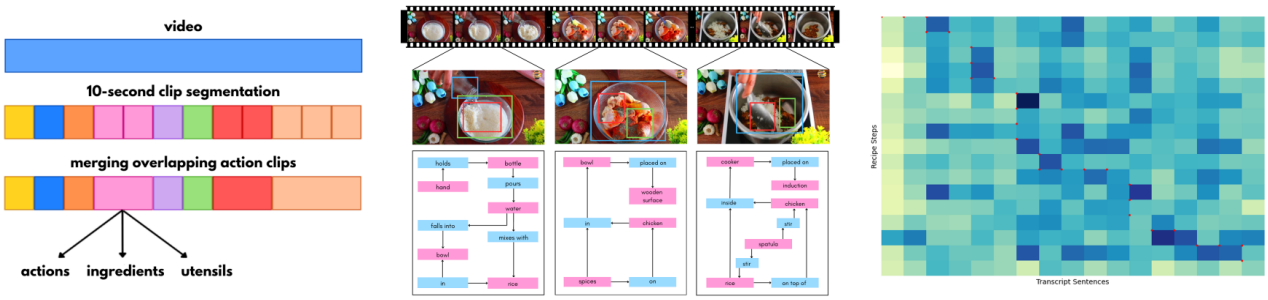
Similarly, the temporal merging process significantly reduced fragmentation in the video segmentation. Across all videos, the number of timestamped clips decreased from 16,761 before merging to 14,479 after merging, representing a 13.6% reduction in segment count while preserving full action coverage.

3.2 Multimodal Alignment: Video, Audio, and Recipe Texts

To build a unified understanding of each *biryani* cooking video, we align three modalities: WhisperX transcripts (temporally ordered narration), InternVL visual segment descriptions (ingredients, utensils, and actions for every 10-second chunk), and manually curated canonical recipes (standard steps and titles per *biryani* type). As shown in Fig 3, alignment begins with coarse filtering, where lowercased and tokenised segment metadata keywords (from detected ingredients/utensils) are matched against transcript lines and recipe steps to eliminate irrelevant pairs. Remaining candidates undergo fine-grained alignment: transcript sentences and recipe steps are embedded with a SentenceTransformer [30], and Dynamic Time Warping (DTW) over cosine distances preserves sequential structure while tolerating omissions, insertions, or reordering—handling

[1] All the prompts used in this paper are available in the supplementary material.

[2] The complete verification workflow is provided in the supplementary material



(a) Video Segmentation Pipeline (b) Video Scene Graphs (c) Recipe-Transcript Alignment Heatmap

Figure 3: Overview of the multimodal video segmentation and alignment pipeline. Panel (a) shows the 10-second clip segmentation of *biryani* cooking videos, where each segment is processed by InternVL-14B to extract visually grounded annotations of actions, ingredients, and utensils. Consecutive segments containing the same action are merged to form continuous spans, improving temporal coherence. Panel (b) presents example video scene graphs depicting detected entities and their relationships. displays an alignment heatmap between canonical recipe steps (vertical) and transcript sentences (horizontal), where colour intensity indicates semantic similarity and the red path represents the optimal sequence alignment computed via Dynamic Time Warping.

deviations from ideal diagonal mappings caused by narration order, granularity mismatches, or pacing differences. For segments passing coarse filtering, we further embed InternVL-extracted actions and recipe steps using BGE [40], compute cosine similarities, assign each chunk to its most semantically relevant recipe step, and rank segments per step with confidence scores. This multimodal alignment enables recipe-aware search, visualisation, and retrieval across heterogeneous time scales and structures.

4 Video Comparison

We aim to understand different *biryani* recipes by comparing their cooking processes. By comparing the cooking process for different types of *biryani*, we can identify common patterns and variations in the cooking methods, ingredients, and techniques used. This can help us understand the unique characteristics of each *biryani* recipe and how they differ.

To compare the cooking processes, including ingredients, methods, actions, etc., different *biryani* varieties (for example, *Hyderabadi biryani* vs *Lucknowi biryani*), we developed a video comparison framework adapted from the VidDiff method [7] that identifies and visualises the differences in cooking actions, ingredients, and techniques. This framework is designed to analyse the cooking videos in our dataset, allowing users to understand how different *biryani* recipes vary in terms of their preparation methods.

To compare the cooking processes across different *biryani* recipes, we adapted the VidDiff framework [7] to our specific use case. The framework consists of three main stages:

Proposer: This stage generates plausible variations for each action class. For each action class, we prompt an LLM to generate plausible ways the action might vary. We also take an action and break it down into sub-actions. Finally, we link the differences to the sub-actions. The LLM is prompted to generate 2-3 variations in the cooking actions that are visually significant and would affect the final frames, and also prompted to generate 2-4 sub-action stages for each action class. The LLM then creates explicit mappings between variations and sub-actions. These mappings specify which differences would be most visually detectable during specific sub-action stages. We employ Qwen2.5 [37].

Frame Retriever: This stage retrieves temporal localisation of sub-actions from cooking videos using CLIP [30]. We embed textual retrieval strings and video frames into a shared semantic space, then compute cosine similarity scores to identify the top-k (k=2) frames that best match each sub-action. This focuses on peak similarity moments where sub-actions are most visually apparent, using ViT-BigG-14 (Open-CLIP) [13].

Action Differentiating: In this final stage, we analyse and visualise the differences between two cooking video segments (segmented by action) using the last stage's localised frames. For each pair of corresponding sub-action segments identified in the previous stage, we pose a multiple-choice question (which were generated from the multiple differences we got from the proposer stage) to a VLM, which determines whether each difference is present in *Video A* or *Video B* or *It's unsure*. We transform our recipe comparison task into a multiple-choice question for the VLM. The VLM is then used to determine which video shows more of the proposed difference, providing a detailed explanation of the observed differences. This allows us to visualise and understand how the cooking processes differ between the two *biryani* recipes. We employ Gemini-2.5-flash-lite [35].

4.1 Results

Our video comparison framework identified meaningful differences across *biryani* varieties. Figure 5 shows that certain cooking stages exhibit minimal variation between *Hyderabadi* and *Lucknowi biryani* , while others display substantial differences.

The framework detected differences in 33.2% of action comparisons. This proportion indicates that while biryani varieties share

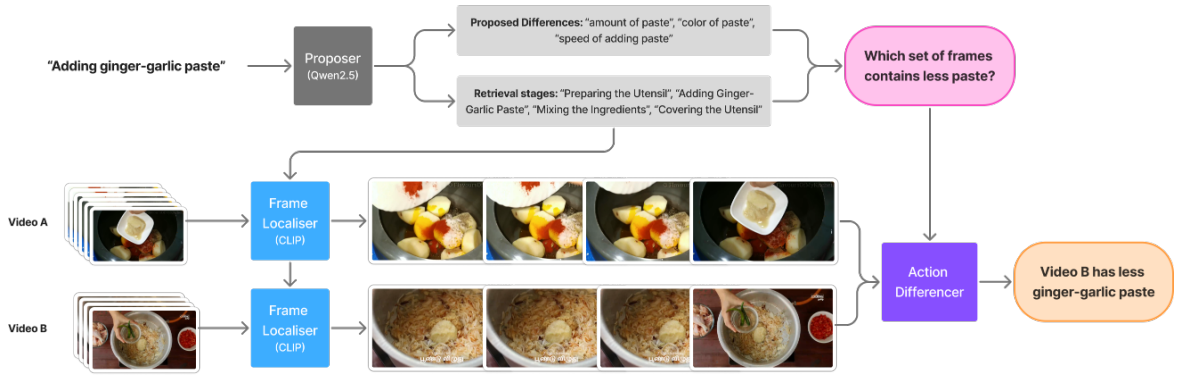


Figure 4: Overview of the video comparison framework for *biryani* recipes. The framework operates through three sequential stages: Proposer (Qwen2-VL) generates plausible variations for each action, Frame Localiser (CLIP) identifies relevant frames, and Action Differencer compares frame pairs to detect differences. This example demonstrates analysis of "Adding ginger-garlic paste," identifying that Video B uses less paste than Video A.

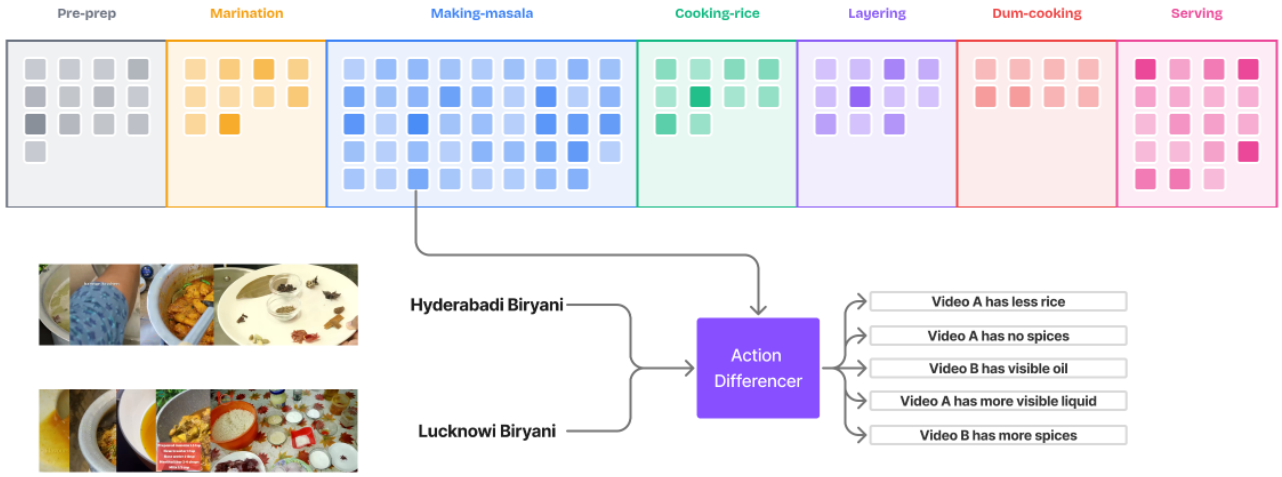


Figure 5: Visualisation of cooking process variations between *Hyderabadi* and *Lucknowi biryani* across several cooking stages. Each coloured section represents a major cooking stage, with individual squares showing specific actions. The opacity of the square is proportional to the degree of variation detected between the two *biryani* styles, where larger squares indicate more significant differences.

Table 1: Distribution of comparison results across randomly paired video segments from 12 *biryani* varieties

Outcome	Percentage of Comparison
Difference detected	33.2%
No detectable difference	66.8%

core cooking procedures, they exhibit distinct variations in execution methods. The detection rate aligns with expectations for regional culinary variants, where fundamental processes remain consistent but specific techniques diverge based on cultural and regional influences.

Further details, visualisations, and discussion of these results are provided in the supplementary material.

To validate the accuracy of the framework, 2000 randomly sampled comparisons were verified by a group of 4–5 independent annotators, with each annotator reviewing a subset of the samples. The verification focused on confirming model-proposed differences rather than performing exhaustive difference detection, which would scale exponentially and is not practically feasible. Table 2 shows accuracy rates across different comparison categories.

Table 2: Manual verification accuracy across categories

Category	Correct	Incorrect
Difference detected	67.5%	32.5%
No difference	45.7%	54.3%

The verification results reveal systematic challenges in the model's performance. The framework achieved 67.5% accuracy for detected differences, indicating reliable identification of actual procedural variations. However, accuracy drops to 45.7% for "no difference" classifications, suggesting the model misses subtle but meaningful variations that human annotators can detect. This performance gap likely stems from the model's limited exposure to Indian cooking contexts during training, resulting in conservative judgments when analysing culturally specific culinary techniques. Additionally, the model occasionally generates false differences or misattributes variations between video clips, highlighting areas for future improvement.

Despite these limitations, the framework successfully captures meaningful procedural differences across regional biryani varieties, providing valuable insights into how traditional cooking methods vary while maintaining cultural authenticity.

5 Video Question Answering

Video Question Answering (VQA) is a key benchmark for evaluating comprehensive scene understanding [14, 31, 39, 42, 44]. Unlike static image tasks, it requires joint reasoning over spatial (object and scene layout), temporal (event ordering, procedural flow), and causal (why actions occur) aspects within and across videos. This capability moves AI/ML systems beyond isolated recognition toward context-aware reasoning in dynamic settings.

In cooking, such reasoning is essential: 'What ingredient was added before the onions?' demands temporal ordering; 'Why was the heat reduced after adding milk?' requires causal inference; and 'Which recipe uses more spices?' involves multi-video comparison. By spanning easy, medium, and hard difficulty tiers, our dataset targets this spectrum—from basic perceptual recognition to complex cross-video reasoning—making it both a challenge for current VLMs and a step toward more general-purpose, reasoning-capable AI.

We construct the dataset using a multi-stage pipeline of temporal segmentation, automated visual description, language model prompting, and manual curation. Difficulty tiers are defined as Easy (single short segment), Medium (entire video comprehension), and Hard (multi-video reasoning). Each video is temporally segmented to capture localised cooking events, with InternVL3-14B [48] producing natural language descriptions of ingredients, utensils, and preparation steps. Gemini-2.0-Flash then integrates these segment-level captions [36] into coherent, visually detailed, step-by-step recipe narratives that comprehensively represent the entire cooking process.

5.1 QA Generation

Easy QA Generation. For easy QA pairs, we focus on individual segments. We randomly sample up to three 10-second segments for each video to generate QA pairs, balancing diversity and computational efficiency. We prompt Llama-3-8B-Instruct [11] to systematically extract three categories of information from each selected segment: (a) ingredients shown (b) utensils used (c) cooking actions performed

To ensure high data quality, we manually review the generated QA pairs for each video, selecting the two most informative and unambiguous examples [3]. This curation step filters out incomplete, repetitive, or low-detail responses, yielding a robust set of easy, segment-grounded QA pairs.

Medium QA Generation. For medium-level QA generation, the goal is to assess the model's comprehension of the entire cooking process in each video, requiring integration of visual and procedural cues across the full temporal span. In contrast to the short-segment focus of easy QA pairs, these questions target broader aspects such as ingredient usage, temporal ordering of key steps, and presentation details. Video summaries are combined with aligned audio transcripts to enable this, providing a rich multimodal textual context that captures visual observations and spoken instructions. Using this input, we prompt Gemini-2.0-Flash [36] to produce a high-level summary and multiple QA pairs, guided by carefully designed question templates tailored to cooking scenarios. These templates emphasize visual elements (e.g., primary ingredients, garnishes, spices), temporal understanding (e.g., sequence of actions, cooking durations, preparation time), and utensil or process details (e.g., vessel type, marination or frying steps), while allowing the model to generate additional contextually relevant questions beyond the provided templates.

Hard QA Generation. For the most challenging QA tier, we evaluate a model's ability to reason across multiple cooking videos, requiring deeper comparative understanding of recipes, cooking styles, and ingredient choices. We first create multimodal summaries of individual videos by combining detailed frame-wise visual descriptions with complete audio transcripts, capturing both rich visual details (ingredients, techniques, utensils, textures, plating) and spoken instructions (quantities, tips, emphases).

We generate hard QA pairs from these summaries by sampling combinations of 2, 3, 4, and 5 videos from the 120-video pool and instructing Gemini-2.5-Flash [35] to analyse their combined content. The model compares, contrasts, and synthesizes details—such as ingredients, cooking methods, spice levels, preparation sequences, and presentation styles—to formulate high-level, reasoning-intensive QAs that require integrating information from multiple sources.

Dataset Statistics. Our QA generation pipeline produces 240 easy, 1,357 medium, and 486 hard question–answer (QA) pairs. The hard QA set is further subdivided based on the number of videos required for reasoning: `hardqa2` (146), `hardqa3` (171), `hardqa4` (82), and `hardqa5` (87). The dataset is evenly split into training and test sets to support model development and evaluation, ensuring balanced representation across all difficulty levels and subsets.

Figure 7 summarizes these statistics. Panel (a) shows the percentage distribution of QA pairs by difficulty, where medium questions dominate (65.1%), followed by hard (23.3%) and easy (11.5%). Panel

[3]Examples of QA pairs for each difficulty tier are provided in the supplementary material

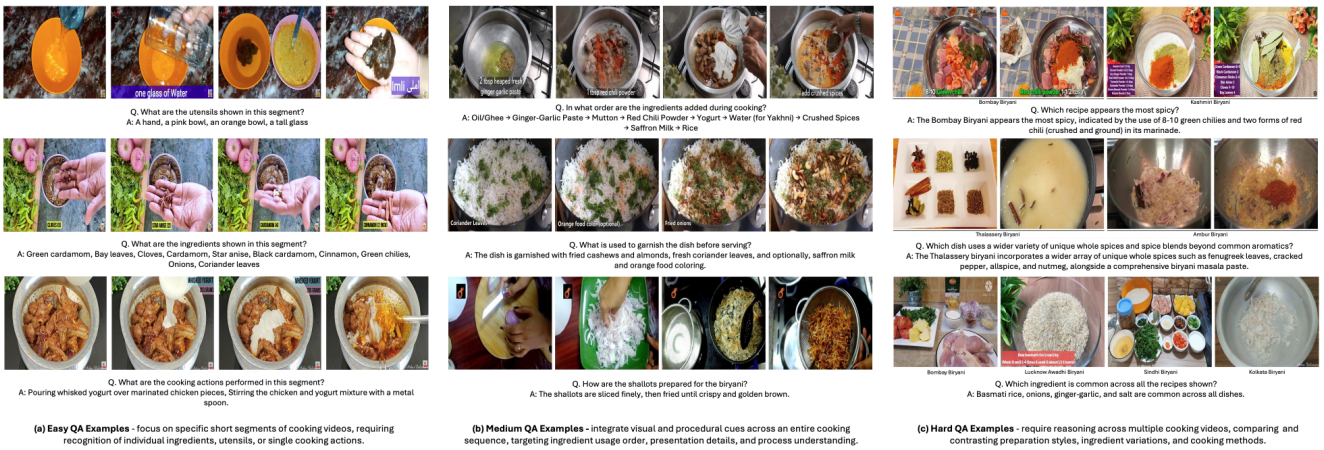


Figure 6: Example QA pairs from the biryani video QA dataset, covering easy, medium, and hard difficulty tiers. Questions were generated via a multi-stage pipeline (temporal segmentation, captioning, summary synthesis, LLM prompting, and human curation). Easy QAs are segment-level recognition tasks, medium QAs require whole-video temporal and procedural understanding, and hard QAs demand multi-video reasoning and comparison. These examples illustrate the dataset's progression from simple perception to complex reasoning.

(b) presents the average answer length for each difficulty type. As expected, harder questions tend to require longer answers, with an average of over 20 words for hard items compared to around 12 words for easy ones. This trend reflects the increased complexity and reasoning demands of higher difficulty levels.

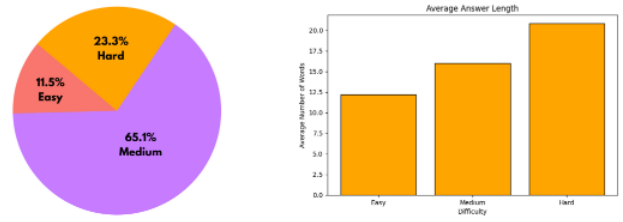


Figure 7: Statistics of the biryani video QA dataset. (a) Distribution of question–answer pairs across difficulty levels. (b) Average answer length per difficulty type, showing a clear upward trend with complexity.

5.2 Results

We benchmark existing video–language models (VLMs) on our QA dataset using both zero-shot and fine-tuned settings. Five open-source VLMs are evaluated in zero-shot mode — InternVL3-8B (internvl3) [48], Qwen2-VL-7B-Instruct (qwen2vl) [38], llava-v1.6-mistral-7b-hf (llavanext) [18], llava-onevision-qwen2-7b-ov-hf (llava ov) [16], and VideoLLaMA3-7B-Image (videollama) [45] — and we fine-tune Llama-3.2-11B-Vision-Instruct (llama3ft) [11] on our dataset with type-specific prompts and frame-sampled inputs to measure domain adaptation gains.

We report standard QA metrics - BLEU, ROUGE-L, and BERTScore - to capture lexical and semantic similarity, but true evaluation lies in the dataset's tier design. The medium and hard tiers deliberately require temporal, procedural, and cross-recipe reasoning, making the tier structure a stronger indicator of reasoning depth than raw metric scores.k

Across all metrics, the fine-tuned Llama-3.2 outperforms zero-shot baselines, with the most significant gains on medium and hard questions. Improvements are most pronounced in BERTScore, indicating stronger semantic alignment in addition to lexical accuracy. Some zero-shot models (e.g., Qwen2-VL, InternVL3) perform competitively in certain tiers, but none match the fine-tuned model's consistency.

For the hard QA tier, we further break down results into `hard2` – `hard5`, corresponding to the number of videos required for reasoning. Tables 3 and 4 present full results. Performance generally declines with more videos, reflecting the difficulty of multi-video reasoning.

We demonstrate a systematic framework for characterising the depth of understanding of AI systems in the cooking domain. Though today's AI systems are very promising for many tasks, there is a good amount of work leftout in developing skills required for understanding fine and specialized skills, as in domains like cooking.

6 Discussions

Application in Skill-Based Video Retrieval. Beyond full-recipe visualisation, our dataset supports targeted instructional search within and across videos. For instance, if a user is interested in understanding how to marinate chicken—a critical step in many *biryani* variants—they can retrieve all video segments across the dataset that involve marination actions. These segments are sourced from different videos but are uniformly timestamped and labelled using our alignment framework. Figure 8 presents an

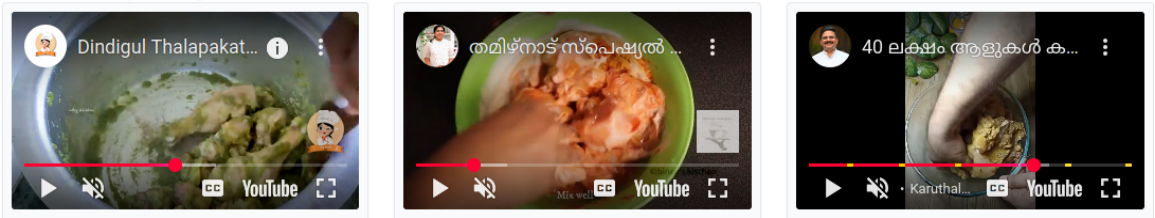


Figure 8: Example of skill-based video retrieval for the query *"marinating chicken"*. The system returns short, timestamped clips from multiple biryani videos where the marination step is visually identified, enabling direct access to semantically relevant moments rather than full unindexed videos.

Table 3: Overall QA performance of VLMs on the QA dataset across easy, medium, and hard difficulty tiers. Best results for each metric–tier combination are highlighted in bold.

VLM	Metric	easy	medium	hard
internvl3	BLEU	0.0294	0.0291	0.0395
	ROUGE-L	0.2184	0.1732	0.2457
	BERTScore	0.1663	0.1628	0.2683
qwen2vl	BLEU	0.0314	0.0209	0.0609
	ROUGE-L	0.1914	0.1189	0.3201
	BERTScore	0.1298	-0.0747	0.3022
llavanext	BLEU	0.0128	0.0216	0.0150
	ROUGE-L	0.1319	0.1367	0.1911
	BERTScore	-0.1732	0.0465	0.0984
llavaov	BLEU	0.0038	0.0278	0.0246
	ROUGE-L	0.0408	0.1383	0.1386
	BERTScore	-0.2586	0.0377	-0.0073
videollama	BLEU	0.0194	0.0787	0.0502
	ROUGE-L	0.1883	0.2713	0.2650
	BERTScore	0.0897	0.3071	0.2445
llama3ft	BLEU	**0.0472**	**0.1683**	**0.1140**
	ROUGE-L	**0.2689**	**0.4214**	**0.4072**
	BERTScore	**0.2660**	**0.4869**	**0.4526**

Table 4: Hard-tier breakdown showing VLM performance on subsets `hard2`, `hard3`, `hard4`, and `hard5`, corresponding to the number of videos required for reasoning.

VLM	Metric	hard2	hard3	hard4	hard5
internvl3	BLEU	0.0432	0.0405	0.0386	0.0322
	ROUGE-L	0.2624	0.2510	0.2444	0.2087
	BERTScore	0.2882	0.2756	0.2532	0.2347
qwen2vl	BLEU	0.0597	0.0679	0.0526	0.0570
	ROUGE-L	0.3300	0.3238	0.3174	0.2990
	BERTScore	0.3107	0.2980	0.3052	0.2932
llavanext	BLEU	0.0052	0.0205	0.0113	0.0239
	ROUGE-L	0.1663	0.2038	0.1718	0.2257
	BERTScore	0.0700	0.1066	0.0727	0.1540
llavaov	BLEU	0.0226	0.0282	0.0215	0.0236
	ROUGE-L	0.1390	0.1459	0.1329	0.1286
	BERTScore	0.0066	0.0094	-0.0400	-0.0327
videollama	BLEU	0.0504	0.0624	0.0339	0.0411
	ROUGE-L	0.2643	0.2870	0.2326	0.2537
	BERTScore	0.2573	0.2552	0.2049	0.2391
llama3ft	BLEU	**0.1073**	**0.1306**	**0.0987**	**0.1068**
	ROUGE-L	**0.4045**	**0.4279**	**0.3845**	**0.3927**
	BERTScore	**0.4622**	**0.4669**	**0.4279**	**0.4319**

example frame retrieved from a marination segment. Unlike traditional video search engines, which return entire videos without pinpointing where the relevant action occurs, our approach enables direct navigation to semantically aligned moments within the video corpus.

Our work opens up many more potential applications in cooking:

- Understanding and documenting the rich cultural heritage of the country. Eventually transferring one to the other in an appropriate manner.
- We hope the deeper video understanding presented here could lead to educational tool and cooking assistants, who can provide contextual assistance with speech and language when integrated with an ego-centric vision.

6.1 Summary

In this work, we presented a systematic computational study of *biryani* preparation videos from across India. We aimed to understand how fine-grained procedural differences manifest in culturally rich cooking practices. We curated the first large-scale *Biryani* Cooking Video Dataset, comprising 120 high-quality YouTube videos spanning 12 distinct regional styles. Building on recent advances in vision–language models (VLMs), we developed a multi-stage framework for temporal segmentation and multimodal alignment between visual content, narration, and canonical recipe text.

We used this aligned representation to introduce a video comparison pipeline that identifies and explains procedural differences between regional variants, enabling interpretable cross-recipe analysis. We further constructed a multi-tier question–answer benchmark to evaluate VLMs on procedural video understanding tasks ranging from localised recognition to multi-video reasoning. Our experiments benchmarked several state-of-the-art VLMs under both zero-shot and fine-tuned settings, highlighting the potential of domain adaptation for structured multimodal reasoning.

Beyond its immediate results, this work provides a foundation for a new class of video understanding benchmarks that combine cultural specificity with fine-grained procedural analysis. The dataset, prompts, and annotations will be released to facilitate reproducibility and further research. Future directions include expanding the scope to other culturally significant dishes, improving alignment robustness in the presence of noisy narration, and developing more efficient VLM prompting strategies for long-form video.

Acknowledgements. We acknowledge and appreciate the support of Google Research / AI in this project.

References

[1] K.T. Achaya. 1994. *Indian Food: A Historical Companion.* Zenodo. doi:10.5281/zenodo.4067897

[2] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774* (2023).

[3] Mahmoud Al-Faris, John Chiverton, David Ndzi, and Ahmed Isam Ahmed. 2020. A review on computer vision-based methods for human action recognition. *Journal of imaging* 6, 6 (2020), 46.

[4] Vishu Antani and Santosh Mahapatra. 2022. Evolution of Indian cuisine: a socio-historical review. *Journal of Ethnic Foods* 9, 1 (2022), 15.

[5] Max Bain, Jaesung Huh, Tengda Han, and Andrew Zisserman. 2023. Whisperx: Time-accurate speech transcription of long-form audio. *arXiv preprint arXiv:2303.00747* (2023).

[6] Praveen Batapati, Duy Tran, Weihua Sheng, Meiqin Liu, and Ruili Zeng. 2014. Video analysis for traffic anomaly detection using support vector machines. In *Proceeding of the 11th World Congress on Intelligent Control and Automation*. IEEE, 5500–5505.

[7] James Burgess, Xiaohan Wang, Yuhui Zhang, Anita Rau, Alejandro Lozano, Lisa Dunlap, Trevor Darrell, and Serena Yeung-Levy. 2025. Video Action Differencing. In *The Thirteenth International Conference on Learning Representations*. https://openreview.net/forum?id=3bcN6xlO6f

[8] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, et al. 2018. Scaling egocentric vision: The epic-kitchens dataset. In *Proceedings of the European conference on computer vision (ECCV)*. 720–736.

[9] Mohamed M Elgamml, Fazly S Abas, and H Ann Goh. 2020. Semantic analysis in soccer videos using support vector machine. *International Journal of Pattern Recognition and Artificial Intelligence* 34, 09 (2020), 2055018.

[10] Junyu Gao and Changsheng Xu. 2021. Learning video moment retrieval without a single annotated video. *IEEE Transactions on Circuits and Systems for Video Technology* 32, 3 (2021), 1646–1657.

[11] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783* (2024).

[12] Matthew Honnibal, Ines Montani, Sofie Van Landeghem, Adriane Boyd, et al. 2020. spaCy: Industrial-strength natural language processing in python. (2020).

[13] Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. 2021. *OpenCLIP*. doi:10.5281/zenodo.5143773 If you use this software, please cite it as below..

[14] Yunseok Jang, Yale Song, Youngjae Yu, Youngjin Kim, and Gunhee Kim. 2017. TGIF-QA: Toward Spatio-Temporal Reasoning in Visual Question Answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[15] Licheng Jiao, Ruohan Zhang, Fang Liu, Shuyuan Yang, Biao Hou, Lingling Li, and Xu Tang. 2021. New generation deep learning for video object detection: A survey. *IEEE Transactions on Neural Networks and Learning Systems* 33, 8 (2021), 3195–3215.

[16] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, et al. 2024. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326* (2024).

[17] Franklin Mingzhe Li, Kaitlyn Ng, Bin Zhu, and Patrick Carrington. 2025. OSCAR: Object Status and Contextual Awareness for Recipes to Support Non-Visual Cooking. In *Proceedings of the Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*. 1–6.

[18] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2024. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 26296–26306.

[19] Cheng Lu, Mark S Drew, and James Au. 2001. Classification of summarized videos using hidden Markov models on compressed chromaticity signatures. In *Proceedings of the ninth ACM international conference on Multimedia*. 479–482.

[20] Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Shahbaz Khan. 2023. Video-chatgpt: Towards detailed video understanding via large vision and language models. *arXiv preprint arXiv:2306.05424* (2023).

[21] Jonathan Malmaud, Jonathan Huang, Vivek Rathod, Nick Johnston, Andrew Rabinovich, and Kevin Murphy. 2015. What's cookin'? interpreting cooking videos using text, speech and vision. *arXiv preprint arXiv:1503.01558* (2015).

[22] Zhang Min-qing and Li Wen-ping. 2021. An automatic classification method of sports teaching video using support vector machine. *Scientific programming* 2021, 1 (2021), 4728584.

[23] Loris Nanni, Stefano Ghidoni, and Sheryl Brahnam. 2017. Handcrafted vs. non-handcrafted features for computer vision classification. *Pattern recognition* 71 (2017), 158–172.

[24] Pradyumna Narayana, J Ross Beveridge, and Bruce A Draper. 2018. Interacting Hidden Markov Models for Video Understanding. *International Journal of Pattern Recognition and Artificial Intelligence* 32, 11 (2018), 1855020.

[25] Eralda Nishani and Betim Çiço. 2017. Computer vision approaches based on deep learning and neural networks: Deep neural networks for video analysis of human pose estimation. In *2017 6th Mediterranean Conference on Embedded Computing (MECO)*. IEEE, 1–4.

[26] Taichi Nishimura, Atsushi Hashimoto, Yoshitaka Ushiku, Hirotaka Kameko, and Shinsuke Mori. 2024. Recipe generation from unsegmented cooking videos. *ACM Transactions on Multimedia Computing, Communications and Applications* (2024).

[27] Marios S Pattichis, Venkatesh Jatla, and Alvaro E ulloa Cerna. 2023. A review of machine learning methods applied to video analysis systems. In *2023 57th Asilomar Conference on Signals, Systems, and Computers*. IEEE, 1161–1165.

[28] Toby Perrett, Ahmad Darkhalil, Saptarshi Sinha, Omar Emara, Sam Pollard, Kranti Kumar Parida, Kaiting Liu, Prajwal Gatti, Siddhant Bansal, Kevin Flanagan, et al. 2025. Hd-epic: A highly-detailed egocentric video dataset. In *Proceedings of the Computer Vision and Pattern Recognition Conference*. 23901–23913.

[29] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2023. Robust speech recognition via large-scale weak supervision. In *International conference on machine learning*. PMLR, 28492–28518.

[30] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics. https://arxiv.org/abs/1908.10084

[31] Xindi Shang, Yicong Li, Junbin Xiao, Wei Ji, and Tat-Seng Chua. 2021. Video visual relation detection via iterative inference. In *Proceedings of the 29th ACM international conference on Multimedia*. 3654–3663.

[32] Vijeta Sharma, Manjari Gupta, Ajai Kumar, and Deepti Mishra. 2021. Video processing using deep learning techniques: A systematic literature review. *IEEE Access* 9 (2021), 139489–139507.

[33] Tulasi Srinivas. 2011. Exploring Indian culture through food. *Education about Asia* 16, 3 (2011), 38–41.

[34] Yunlong Tang, Jing Bi, Siting Xu, Luchuan Song, Susan Liang, Teng Wang, Daoan Zhang, Jie An, Jingyang Lin, Rongyi Zhu, et al. 2025. Video understanding with large language models: A survey. *IEEE Transactions on Circuits and Systems for Video Technology* (2025).

[35] Gemini Team. 2025. Gemini 2.5: Pushing the Frontier with Advanced Reasoning, Multimodality, Long Context, and Next Generation Agentic Capabilities. *arXiv preprint arXiv:2507.06261* (July 2025).

[36] Gemini Robotics Team, Saminda Abeyruwan, Joshua Ainslie, Jean-Baptiste Alayrac, Montserrat Gonzalez Arenas, Travis Armstrong, Ashwin Balakrishna, Robert Baruch, Maria Bauza, Michiel Blokzijl, et al. 2025. Gemini robotics: Bringing ai into the physical world. *arXiv preprint arXiv:2503.20020* (2025).

[37] Qwen Team. 2024. Qwen2.5: A Party of Foundation Models. https://qwenlm.github.io/blog/qwen2.5/

[38] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. 2024. Qwen2-VL: Enhancing Vision-Language Model's Perception of the World at Any Resolution. *arXiv preprint arXiv:2409.12191* (2024).

[39] Junbin Xiao, Xindi Shang, Angela Yao, and Tat-Seng Chua. 2021. NExT-QA: Next Phase of Question-Answering to Explaining Temporal Actions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 9777–9786.

[40] Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. 2023. C-Pack: Packaged Resources To Advance General Chinese Embedding. arXiv:2309.07597 [cs.CL]

[41] Lexing Xie, Shih-Fu Chang, Ajay Divakaran, and Huifang Sun. 2002. Learning hierarchical hidden Markov models for video structure discovery. *ADVENT Group, Columbia Univ., New York, Tech. Rep* 6 (2002).

[42] Dejing Xu, Zhou Zhao, Jun Xiao, Fei Wu, Hanwang Zhang, Xiangnan He, and Yueting Zhuang. 2017. Video question answering via gradually refined attention over appearance and motion. In *Proceedings of the 25th ACM international conference on Multimedia*. 1645–1653.

[43] Frank F Xu, Lei Ji, Botian Shi, Junyi Du, Graham Neubig, Yonatan Bisk, and Nan Duan. 2020. A benchmark for structured procedural knowledge extraction from cooking videos. *arXiv preprint arXiv:2005.00706* (2020).

[44] Pinci Yang, Xin Wang, Xuguang Duan, Hong Chen, Runze Hou, Cong Jin, and Wenwu Zhu. 2022. Avqa: A dataset for audio-visual question answering on videos. In *Proceedings of the 30th ACM international conference on multimedia*. 3480–3491.

[45] Boqiang Zhang, Kehan Li, Zesen Cheng, Zhiqiang Hu, Yuqian Yuan, Guanzheng Chen, Sicong Leng, Yuming Jiang, Hang Zhang, Xin Li, et al. 2025. Videollama 3: Frontier multimodal foundation models for image and video understanding. *arXiv preprint arXiv:2501.13106* (2025).

[46] Luowei Zhou, Chenliang Xu, and Jason Corso. 2018. Towards automatic learning of procedures from web instructional videos. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 32.

[47] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. 2023. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592* (2023).

[48] Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, et al. 2025. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479* (2025).