PEER-REVIEWED JOURNAL OF COMPUTER SCIENCE (PRJCS)

ISSN Online: 3139-5082 Open Access Scholarly Journal

Multimodal Large Language Models: A Survey of Vision-Language Architectures

M. Keerthika

Assistant Professor, Department of Computer Science, Yuvakshetra Institute of Management Studies, Palakkad, India.

Article information

Received: 5th June 2026
Received in revised form: 24th June 2026
Accepted: 30th June 2026
Available online: 9th July 2026

Volume: 1
Issue: 7
DOI: https://doi.org/10.5281/zenodo.21258461

Abstract

Multimodal large language models, often called vision-language models, extend text-only language models so that a single system can read images together with words and produce grounded natural-language responses. This survey gives a structured and honest account of how these systems are built, trained, and measured. It first sets out the background that made them possible: the transformer sequence model, large pretrained language models, the vision transformer that treats an image as a sequence of patches, and contrastive image-text pretraining that aligns pictures and captions in a shared space. It then organises the design space into three parts, namely the vision encoder, the cross-modal connector that maps visual features into the language model, and the language decoder, and it contrasts query-based connectors, linear projectors, and gated cross-attention as well as early and late fusion. Representative systems including CLIP, Flamingo, BLIP-2, LLaVA, GPT-4V, Gemini, Qwen-VL, and PaliGemma are described from public reports, with proprietary details marked as undisclosed. The survey reviews training stages, benchmark families for visual question answering, document and chart reading, expert reasoning, and grounding, and it treats object hallucination and evaluation contamination as first-class problems. Reported scores are presented only as rounded representative illustrations drawn from heterogeneous sources. The article closes with open challenges in spatial reasoning, faithful evaluation, efficiency, and safety, and with directions for any-to-any and video-capable models.

.

Keywords: - Vision-Language Models, Multimodal Large Language Models, Contrastive Pretraining, Instruction Tuning, Visual Question Answering, Hallucination, Benchmarks, Cross-Modal Fusion.

.

I. INTRODUCTION

The transformer architecture reshaped how machines process sequences by replacing recurrence with self-attention, so that every token can attend to every other token in parallel [1]. Scaling this design to very large text corpora produced language models that follow instructions, answer questions, and write code with a single set of weights [2]. These systems remain blind, however, because they read only words. A photograph, a scanned invoice, a medical scan, a chart, or a screenshot cannot be described to them without a human first turning the pixels into text. Multimodal large language models, widely referred to as vision-language models, remove that bottleneck by letting one model read images and text together and reply in natural language.

A vision-language model, as used in this survey, is a system that couples a visual encoder with a language model so that image content is injected into the token stream the language model consumes. The language model still generates text one token at a time, but its context now contains a representation of what an image shows. This simple idea supports a wide span of behaviour: describing a scene, reading text inside a photo, answering a question about a diagram, comparing two pictures, or pointing to a region that matches a phrase. The appeal is a general interface. Rather than

training a separate network for captioning, another for visual question answering, and another for document reading, one instruction-following model addresses many tasks through prompts.

The motivation for a single multimodal interface is both practical and scientific. On the practical side, one model that reads pictures and text can serve document processing, accessibility for people with low vision, tutoring over diagrams, medical and industrial inspection, and assistants that see a user screen. On the scientific side, grounding language in perception tests whether a model has learned meaning that connects words to the visual world rather than only statistical regularities of text. A caption generator that has never seen the object it names, or a question answerer that relies on language priors instead of the image, reveals that the grounding is shallow. Studying how, when, and why vision-language models fail is therefore as informative as celebrating what they achieve.

This article surveys how such models are constructed, trained, and evaluated, and it aims to stay honest about what is known and what is not. Many of the strongest systems are proprietary, and their exact encoders, data, and parameter counts are not public. Where a detail has not been disclosed, this survey says so rather than guessing. Where numbers appear, they are rounded representative values gathered from heterogeneous public reports and leaderboards, and they are not directly comparable across models because prompting, decoding, and evaluation protocols differ. The goal is a clear map of the design space and the open problems, not a ranking. The remaining sections cover background, architectures, representative models, training paradigms, capabilities and benchmarks, extensions and efficiency, challenges and future directions, and a conclusion.

II. BACKGROUND

A. Language Models and the Transformer

Two lines of work prepared the ground for vision-language models. The first is the transformer language model. Masked encoders such as BERT showed that pretraining on large unlabelled text and then fine-tuning transfers well across many language tasks [3]. Decoder-only models trained to predict the next token showed that a single model, given a natural-language prompt and a few examples, can perform tasks it was never explicitly trained for [2]. This in-context ability is central to vision-language models, because it means a frozen or lightly adapted language model can be repurposed to describe or reason about images once visual tokens are supplied.

B. Vision Transformers

The second line is the vision transformer. Rather than processing an image with convolutions, the vision transformer splits the picture into fixed-size patches, embeds each patch as a token, adds position information, and passes the sequence through the same self-attention blocks used for text [4]. Trained at sufficient scale, it matches or surpasses convolutional networks on image classification. Its importance for vision-language models is that it turns an image into a sequence of tokens whose format already resembles what a language transformer expects, which makes joining the two modalities more natural.

C. Contrastive Image-Text Pretraining

The bridge between the two modalities is contrastive pretraining. CLIP trains an image encoder and a text encoder jointly on hundreds of millions of image-caption pairs collected from the web, pulling matching image and text embeddings together and pushing mismatched pairs apart [5]. The result is a shared embedding space where a picture of a dog and the words describing it land near each other. This yields strong zero-shot classification by comparing an image to text prompts, and, more importantly for this survey, it produces visual features that already carry semantic, language-aligned meaning. Many later systems reuse a CLIP-style encoder precisely because its features are easy for a language model to interpret. Early bridging systems such as Flamingo demonstrated that a small number of trainable components could connect a powerful frozen vision encoder to a frozen language model for few-shot multimodal tasks [6]. BLIP complemented this line by showing that bootstrapped captioning and filtering can clean noisy web pairs before they are used for pretraining, which improves both understanding and generation [7].

D. Scale and Emergent Behaviour

One reason these ingredients combine well is that both the language and vision components benefit from scale. Larger language models trained on more text acquire in-context learning and broader world knowledge, and larger contrastive encoders trained on more image-text pairs produce cleaner, more transferable features [2], [5]. When a capable language model is joined to a capable encoder, the multimodal system inherits both. This is also why proprietary systems that do not disclose their scale can still be characterised by behaviour: their reports describe what the model does even when they withhold how it was built [8], [9]. Scale is not the whole story, however, because the choice of connector, the resolution of the encoder, and the quality of instruction data all shape the final model as strongly as parameter count.

III. ARCHITECTURES OF VISION-LANGUAGE MODELS

Most vision-language models share a three-part anatomy: a vision encoder that turns pixels into features, a cross-modal connector that maps those features into the space the language model reads, and a language decoder that generates the response. The main design decisions concern which encoder to use, how the connector is built, and whether the language model is frozen, partly adapted, or fully trained. The connector is where designs differ most, and it is the component that determines how many visual tokens enter the context and how faithfully visual detail survives the transfer.

A. Vision Encoders

The encoder is usually a vision transformer pretrained by contrastive or supervised objectives, often a CLIP-style or sibling encoder whose features are language-aligned [4], [5]. Higher input resolution and finer patching help with small text and dense charts but increase the number of visual tokens and the compute cost. Some systems keep the encoder frozen to preserve its general features and reduce training cost, while others fine-tune it so that visual features adapt to the downstream language model.

B. Cross-Modal Connectors

The connector maps encoder outputs into tokens the language model can attend to. Three families are common. The first is the linear or shallow projector, used by LLaVA, which applies a small learned mapping so that each visual feature becomes a token placed directly in the language model input [10]. This design is simple and preserves spatial detail but produces many tokens. The second is the query-based connector, exemplified by the Querying Transformer, or Q-Former, in BLIP-2 [11]. A fixed set of learnable query vectors attends to the frozen image features and compresses them into a short sequence of visual tokens, which sharply reduces context length while keeping the encoder and language model frozen. The third is gated cross-attention, introduced in Flamingo, where new cross-attention layers are inserted between the frozen language layers so the text stream can read image features without replacing the language weights [6]. Query-based and cross-attention connectors trade some spatial fidelity for efficiency and stable training on top of frozen backbones.

C. Early Versus Late Fusion

Fusion describes when the two modalities meet. In late fusion, each modality is encoded largely on its own and combined near the output, which is typical of contrastive models such as CLIP that compare a single image vector with a single text vector [5]. In early fusion, visual tokens enter the language model near its input and are processed jointly with text through the full depth of the network, which is typical of instruction-tuned generators such as LLaVA and BLIP-2 [11], [10]. Early fusion supports open-ended dialogue and fine-grained reasoning because visual and textual information interact at every layer, whereas late fusion is efficient for retrieval and classification. Table 1 summarises representative architectures and marks proprietary details that have not been disclosed.

D. Resolution and the Token Budget

A cross-cutting decision is how many visual tokens the language model must read. A single image at modest resolution can already turn into hundreds of tokens, and high resolution or tiled inputs push this into the thousands, which raises memory use and slows generation. This tension explains much of the diversity in connector design. A linear projector keeps one token per visual feature and preserves detail, so it reads dense charts and small text well, but it is expensive [10]. A query-based connector fixes a small token budget regardless of image size, which is cheap and stable but can drop fine detail [11]. Practical systems therefore balance resolution against token count, sometimes using tiling with a shared encoder to keep detail where it matters while capping the total context length.

Table 1. Representative Vision-Language Architectures (Public Reports; Undisclosed Items Marked)

Model	Year	Vision encoder	Connector	Language model	Fusion
CLIP	2021	ViT / ResNet	Shared embedding	Text encoder	Late (contrastive)
Flamingo	2022	Frozen NFNet vision	Gated cross-attention	Frozen Chinchilla-style	Early (cross-attn)
BLIP-2	2023	Frozen ViT	Q-Former	Frozen OPT / FlanT5	Early
LLaVA	2023	CLIP ViT-L/14	Linear / MLP projector	Vicuna (LLaMA)	Early
Qwen-VL	2023	ViT (OpenCLIP init)	Cross-attn adapter	Qwen LLM	Early
PaliGemma	2024	SigLIP-So400m	Linear projector	Gemma-2B	Early
GPT-4V	2023	Not disclosed	Not disclosed	GPT-4	Not disclosed
Gemini	2023	Not disclosed	Not disclosed	Not disclosed	Natively multimodal

IV. REPRESENTATIVE MODELS

This section describes widely cited systems from public reports. Open models disclose their components and often their weights, while several frontier models disclose behaviour and evaluations but not architecture or data.

A. Contrastive and Early Bridging Models

CLIP is the reference contrastive model and remains a common source of language-aligned visual features [5]. Flamingo showed that gated cross-attention over frozen vision and language backbones supports few-shot learning from interleaved image-text sequences, so a handful of examples in the prompt can steer new tasks [6]. BLIP unified image-text understanding and generation with a bootstrapped captioning and filtering pipeline that cleans noisy web data [7]. BLIP-2 then introduced the Q-Former to connect a frozen image encoder to a frozen language model efficiently, which lowered training cost and popularised the query-based connector [11].

B. Instruction-Tuned Open Models

LLaVA demonstrated that a simple linear projector plus visual instruction tuning, using instruction data generated with a language-only model, yields a capable open assistant for visual dialogue [10]. InstructBLIP built instruction tuning on the BLIP-2 backbone with an instruction-aware Q-Former and a broad task mixture [12]. Qwen-VL added higher-resolution inputs, reading of text in images, and region grounding while keeping an open release [13]. PaliGemma paired a SigLIP vision encoder with a compact Gemma language model to give a small, versatile base model designed for transfer to many downstream tasks [14]. These open systems make the design space reproducible and are common baselines in research.

C. Frontier Proprietary Models

GPT-4 with vision, described in its technical report, accepts image and text prompts and performs strongly on many academic and professional tasks, though the report explicitly withholds architecture, data, and training details for competitive and safety reasons [8]. Later releases in the same family extended interaction to real-time voice and richer image handling, again with limited public architectural detail. Gemini was presented as a family of natively multimodal models trained across text, image, audio, and video, with the largest variant reported to advance the state of the art on many benchmarks [9]. Because these systems do not disclose their encoders or connectors, Table 1 marks those fields as not disclosed, and any comparison against open models should be read with that asymmetry in mind.

V. TRAINING PARADIGMS

Training a vision-language model usually proceeds in stages, and different systems emphasise different stages. Recent surveys organise the space along these lines and note that most capable models combine several objectives rather than relying on one [15].

A. Contrastive Alignment

The first paradigm is contrastive alignment, which learns a shared image-text space from large web pairs by matching correct pairs and separating incorrect ones [5]. This stage does not by itself produce a conversational model, but it produces visual features and, in dual-encoder form, a strong retrieval and zero-shot classifier. Many generative systems reuse a contrastively trained encoder as their eyes.

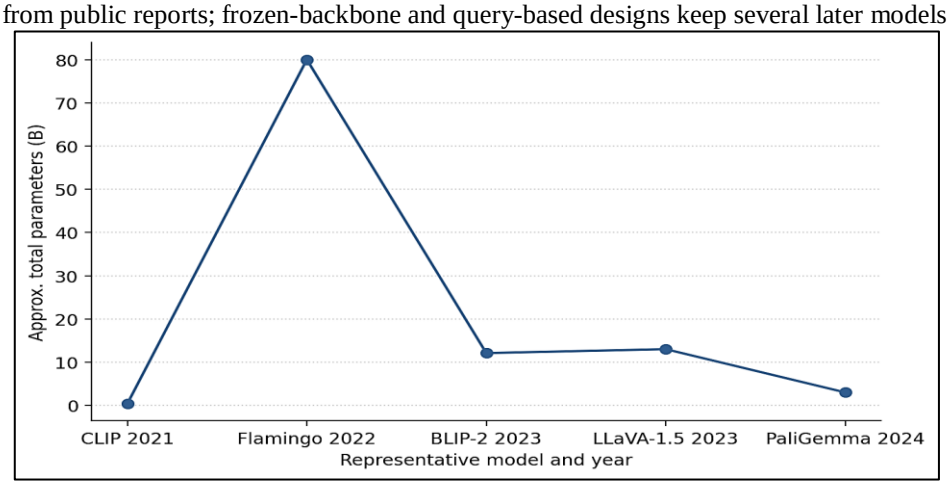
B. Generative Pretraining and Connector Alignment

The second paradigm trains the connector, and sometimes the language model, to generate text conditioned on images. A common recipe first aligns the connector on image-caption data while keeping the backbones frozen, so the projected visual tokens land in a region the language model can read, and then unfreezes more of the network for richer generation [11], [10]. Bootstrapped data cleaning improves the noisy captions that dominate web collections [7].

C. Instruction Tuning and Alignment

The third paradigm is visual instruction tuning, which trains the model on prompts paired with desired responses so it follows open-ended requests, holds multi-turn dialogue, and reasons about images rather than only captioning them [10], [12]. A further alignment step tunes the model toward helpful and safe behaviour using human or model preference signals, which reduces unsafe or off-task outputs. Figure 1 sketches how representative model scale has not increased monotonically over time, because efficient connectors and frozen backbones allowed smaller systems to remain competitive.

Fig. 1: Approximate total parameter count of representative vision-language systems by release (rounded representative values from public reports; frozen-backbone and query-based designs keep several later models small).



Data quality often matters more than the exact objective. The instruction data used by LLaVA was generated by prompting a language-only model to write diverse questions and answers about images whose contents were described in text, which produced varied conversational, descriptive, and reasoning examples at low cost [10]. InstructBLIP mixed

many existing datasets cast into an instruction format so the model would generalise across task types [12]. The recurring lesson is that a modest model trained on clean, varied, instruction-style data can outperform a larger model trained on narrow or noisy data, which is why bootstrapped and filtered corpora recur across the field [7]. Because instruction and alignment data are rarely released in full for proprietary systems, the exact recipe behind frontier behaviour usually stays private [8].

VI. CAPABILITIES, BENCHMARKS, AND EVALUATION

Vision-language models are measured on a growing family of benchmarks, each stressing a different skill. This section reviews the main task types and the datasets used to score them, then presents representative reported numbers with the caution that they are not directly comparable.

A. Visual Question Answering and Captioning

Visual question answering asks a natural-language question about an image and expects a natural-language answer. The original VQA dataset framed this task and exposed how strong language priors can let a model guess answers without truly looking [16]. VQAv2 rebalanced the data with complementary image pairs so that the same question has different answers for different images, which reduces that shortcut [17]. Image captioning, standardised on collections such as Microsoft COCO, measures whether a model can produce a fluent and accurate description of a scene [18]. These tasks remain useful sanity checks even as harder benchmarks appear.

B. Document, Chart, and Text-in-Image Understanding

Many real images contain text and structured graphics. TextVQA requires reading words inside photographs to answer questions, which stresses optical reading rather than only object recognition [19]. ChartQA asks questions about bar, line, and pie charts and demands both visual parsing and arithmetic or logical reasoning over the extracted values [20]. Document and chart understanding are where higher input resolution matters most, and where weak encoders or aggressive token compression tend to fail.

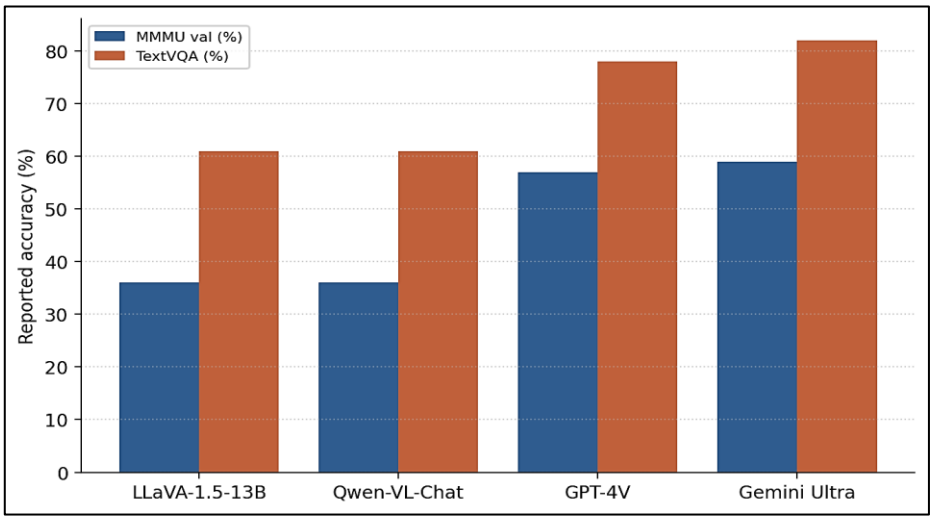
C. Expert Reasoning and Grounding

MMMU raises the difficulty to college-level, multi-discipline questions that mix diagrams, tables, and domain knowledge across subjects such as science, medicine, and engineering, and it remains challenging even for frontier systems [21]. Grounding tasks ask a model to localise the region that matches a phrase, connecting language to spatial position rather than to the whole image. Table 2 lists representative benchmark suites and their focus, and Figure 2 shows rounded representative scores that should be read as illustrations of trend, not as an exact ranking.

Table 2. Representative Benchmark Suites for Vision-Language Models

Benchmark	Year	Primary focus	Approx. size	Typical metric
VQAv2	2017	General visual QA	1.1M questions	Accuracy
TextVQA	2019	Reading text in images	45K questions	Accuracy
ChartQA	2022	Chart reasoning	32K QA pairs	Relaxed accuracy
MMMU	2024	Expert multi-discipline	11.5K questions	Accuracy
POPE	2023	Object hallucination	Probing set	Accuracy / F1
MME	2023	Broad perception+cognition	14 subtasks	Score
MMBench	2024	Fine-grained ability audit	Multiple axes	Accuracy

Fig. 2: Reported representative accuracy of several systems on two benchmarks (rounded values compiled from heterogeneous public reports; prompting and evaluation protocols differ, so bars illustrate trend and are not directly comparable).



The scores above show a consistent qualitative pattern, namely that proprietary frontier systems tend to lead on expert reasoning while strong open models stay competitive on reading and perception. They also show why single numbers mislead: TextVQA and MMMU stress different skills, and a model that reads text well may still struggle with multi-step reasoning. General-purpose audits such as MME and MMBench were built to break scores into finer abilities so that one aggregate figure does not hide specific weaknesses [22], [23].

D. Metrics and Their Limits

The metrics themselves deserve scrutiny. Classic captioning scores reward overlap with reference sentences and can miss whether a description is actually faithful to the image. Visual question answering accuracy depends on how answers are normalised and matched, so a correct answer phrased differently from the reference can be marked wrong. Multiple-choice formats, used by many recent suites, are convenient and stable but let a model guess and can reward pattern matching over genuine understanding. Free-form generation is more realistic but harder to score, and using a strong model as an automatic judge introduces its own biases. No single metric captures competence, which is why this survey treats the reported numbers in Figure 2 as indicators of trend and recommends reading several complementary benchmarks together [21], [22], [23].

VII. EXTENSIONS, HALLUCINATION, AND EFFICIENCY

A. Video, Audio, and Any-to-Any

The image-plus-text template extends naturally to more modalities. Video is treated as a sequence of frames, which multiplies the number of visual tokens and raises the need for temporal modelling and token reduction. Audio can be encoded and injected in the same way as image tokens, letting a model transcribe, describe, or reason over sound. Systems trained across text, image, audio, and video from the start are described as natively multimodal, and some newer models also generate images or speech, moving toward any-to-any behaviour where inputs and outputs span several modalities [9]. These extensions inherit the connector and fusion choices discussed earlier and add the problem of aligning very different token rates.

Video sharpens the efficiency problem more than any other modality, because a short clip sampled at even a few frames per second yields far more visual tokens than a single image. Practical video models therefore sample frames sparsely, pool tokens across time, or use query-based compression so the language context stays affordable, at the cost of missing fast events between sampled frames [11]. Temporal grounding, meaning locating when something happens rather than only where, is an added demand that image benchmarks do not test. Audio brings its own alignment challenge, since speech unfolds at a different rate from vision, and reasoning that combines what is seen with what is heard requires the model to fuse streams that were tokenised on different clocks. Any-to-any systems, which both read and produce multiple modalities, are the most general form of this design and the least mature.

B. Hallucination and Faithful Evaluation

A persistent failure is hallucination, where the model states things that are not in the image, such as naming objects that are absent. POPE measures object hallucination by asking simple yes or no questions about whether specific objects appear, using random, popular, and adversarial object choices to expose bias toward frequently co-occurring items [24]. Hallucination matters because a fluent but wrong answer can be more harmful than an obvious error, and because standard accuracy metrics do not always catch it. Faithful evaluation therefore combines targeted probes like POPE with broad audits like MME and MMBench rather than trusting a single leaderboard number [24], [22], [23].

C. Efficiency

Efficiency is a practical constraint because visual tokens dominate context length, especially at high resolution or for video. Query-based connectors compress many image features into a short sequence, which is one reason the Q-Former design became popular [11]. Freezing the vision encoder and language model and training only a small connector cuts training cost and memory. Compact base models such as PaliGemma show that a small language model with a good encoder and projector can transfer widely, which supports on-device and cost-sensitive deployment [14]. The broad capabilities and ethical challenges of large language models that these systems inherit have themselves been surveyed [25], and the parallel effort on small language models for on-device and private intelligence directly motivates compact multimodal designs whose language backbone can run within tight memory budgets [26]. Figure 1 reflected this point: later systems are not always larger, because architecture and training choices can substitute for raw scale.

VIII. CHALLENGES AND FUTURE DIRECTIONS

Despite rapid progress, several problems are unresolved and shape the research agenda. Recent surveys catalogue many of them and caution against reading benchmark gains as evidence of general visual competence [15].

A. Spatial and Fine-Grained Reasoning

Spatial reasoning remains weak. Models often misjudge relative position, counting, orientation, and precise localisation, in part because contrastive encoders reward global semantic matching over exact geometry, and in part because connectors that compress visual tokens discard spatial detail [5], [11]. Reading dense documents and small text also stresses resolution and token budgets. Improving these skills without exploding context length is an active direction.

B. Hallucination and Evaluation Contamination

Hallucination persists even in strong models and grows in long, open-ended generation [24]. A second and quieter risk is evaluation contamination, where benchmark images or questions leak into training data, so reported scores overstate genuine ability. Because many training corpora are web-scale and not fully documented, contamination is hard to rule out, which strengthens the case for held-out, freshly collected, and probing-style evaluations rather than static public test sets [21], [22].

C. Safety and Transparency

Safety adds a visual attack surface: harmful or manipulative content can be embedded in images, and instructions can be hidden in a picture to steer a model. Alignment tuning reduces but does not remove such risks. Transparency is a related concern, since the strongest systems disclose little about architecture and data, which limits reproducibility and independent auditing [8], [9]. Honest reporting, documented data, and shared evaluation protocols would help the field compare systems fairly.

D. Multilinguality, Culture, and Bias

Because training data is dominated by English text and by images drawn from a limited set of sources, models often perform better in English and on scenes common in that data, and worse on other languages, scripts, and cultural contexts. The same imbalance can encode social bias, so a model may make stereotyped associations from a face or a setting. Benchmarks are affected too, since a suite written in one language and drawn from one region measures a narrow slice of the world [15], [21]. Broadening data coverage, building evaluation in more languages, and auditing for biased associations are necessary for systems meant to serve a global user base rather than a single locale.

E. Future Directions

Promising directions include stronger and higher-resolution encoders that preserve spatial structure, connectors that balance token compression against fidelity, unified any-to-any models that read and generate across modalities, and grounding that ties language to precise regions and to actions in the world. Better evaluation is equally important: dynamic benchmarks, contamination checks, and calibrated uncertainty so that a model can say when it is unsure rather than confidently hallucinate. Progress on these fronts would move vision-language models from strong demonstration systems toward dependable tools.

IX. CONCLUSION

Vision-language models arose from a clear recipe: take a capable language model, add a vision encoder, and connect the two so images become tokens the language model can read. The transformer, large pretrained language models, the vision transformer, and contrastive image-text pretraining supplied the pieces, and query-based connectors, linear projectors, and gated cross-attention supplied the joins. Open systems such as BLIP-2, LLaVA, InstructBLIP, Qwen-VL, and PaliGemma made the design space reproducible, while proprietary systems such as GPT-4V and Gemini raised the ceiling but disclosed little about how. Benchmarks for visual question answering, document and chart reading, expert reasoning, and grounding chart the progress, yet reported scores must be read as heterogeneous illustrations rather than exact rankings.

This survey has stressed honesty over hype. The models are genuinely useful and improving quickly, but they still hallucinate, reason poorly about space, depend on evaluation that may be contaminated, and often hide their internals. The most valuable next steps are therefore as much about measurement and transparency as about architecture: faithful, contamination-resistant evaluation, calibrated uncertainty, documented data, and connectors and encoders that keep spatial detail without inflating cost. With those in place, vision-language models can grow from impressive general interfaces into trustworthy assistants for reading and reasoning about the visual world [15].

REFERENCES

[1] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutierrez, “Recommender systems survey,” Knowledge-Based Systems, vol. 46, pp. 109–132, 2013.

[2] F. Ricci, L. Rokach, and B. Shapira, Eds., Recommender Systems Handbook, 2nd ed. New York, NY, USA: Springer, 2015.

[3] N. Manouselis, H. Drachsler, R. Vuorikari, H. Hummel, and R. Koper, “Recommender systems in technology enhanced learning,” in Recommender Systems Handbook, F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, Eds. Boston, MA, USA: Springer, 2011, pp. 387–415.

[4] H. Drachsler, K. Verbert, O. C. Santos, and N. Manouselis, “Panorama of recommender systems to support learning,” in Recommender Systems Handbook, 2nd ed., F. Ricci, L. Rokach, and B. Shapira, Eds. New York, NY, USA: Springer, 2015, pp. 421–451.

[5] J. K. Tarus, Z. Niu, and G. Mustafa, “Knowledge-based recommendation: A review of ontology-based recommender systems for e-learning,” Artificial Intelligence Review, vol. 50, no. 1, pp. 21–48, 2018.

[6] G. George and A. M. Lal, “Review of ontology-based recommender systems in e-learning,” Computers & Education, vol. 142, Art. no. 103642, 2019.

[7] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, “Item-based collaborative filtering recommendation algorithms,” in Proc. 10th Int. Conf. World Wide Web (WWW), Hong Kong, 2001, pp. 285–295.

[8] Y. Koren, R. Bell, and C. Volinsky, “Matrix factorization techniques for recommender systems,” Computer, vol. 42, no. 8, pp. 30–37, 2009.

[9] P. Lops, M. de Gemmis, and G. Semeraro, “Content-based recommender systems: State of the art and trends,” in Recommender Systems Handbook, F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, Eds. Boston, MA, USA: Springer, 2011, pp. 73–105.

[10] R. Burke, “Hybrid recommender systems: Survey and experiments,” User Modeling and User-Adapted Interaction, vol. 12, no. 4, pp. 331–370, 2002.

[11] S. Zhang, L. Yao, A. Sun, and Y. Tay, “Deep learning based recommender system: A survey and new perspectives,” ACM Computing Surveys, vol. 52, no. 1, pp. 1–38, 2019.

[12] T. R. Gruber, “A translation approach to portable ontology specifications,” Knowledge Acquisition, vol. 5, no. 2, pp. 199–220, 1993.

[13] R. Studer, V. R. Benjamins, and D. Fensel, “Knowledge engineering: Principles and methods,” Data & Knowledge Engineering, vol. 25, nos. 1–2, pp. 161–197, 1998.

[14] T. Berners-Lee, J. Hendler, and O. Lassila, “The Semantic Web,” Scientific American, vol. 284, no. 5, pp. 34–43, 2001.

[15] IEEE Standard for Learning Object Metadata, IEEE Std 1484.12.1-2002, IEEE, 2002.

[16] A. Klasnja-Milicevic, B. Vesin, M. Ivanovic, and Z. Budimac, “E-learning personalization based on hybrid recommendation strategy and learning style identification,” Computers & Education, vol. 56, no. 3, pp. 885–899, 2011.

[17] M. Salehi and I. N. Kamalabadi, “Hybrid recommendation approach for learning material based on sequential pattern of the accessed material and the learner’s preference tree,” Knowledge-Based Systems, vol. 48, pp. 57–69, 2013.

[18] J. K. Tarus, Z. Niu, and A. Yousif, “A hybrid knowledge-based recommender system for e-learning based on ontology and sequential pattern mining,” Future Generation Computer Systems, vol. 72, pp. 37–48, 2017.

[19] S. Shishehchi, S. Y. Banihashem, and N. A. M. Zin, “A proposed semantic recommendation system for e-learning: A rule and ontology based e-learning recommendation system,” in Proc. Int. Symp. Information Technology (ITSim), Kuala Lumpur, Malaysia, 2010, pp. 1–5.

[20] S. Shanshan, G. Mingjin, and L. Lijuan, “An improved hybrid ontology-based approach for online learning resource recommendations,” Educational Technology Research and Development, vol. 69, no. 5, pp. 2637–2661, 2021.

[21] A. Hogan et al., “Knowledge graphs,” ACM Computing Surveys, vol. 54, no. 4, pp. 1–37, 2021.

[22] Q. Guo et al., “A survey on knowledge graph-based recommender systems,” IEEE Transactions on Knowledge and Data Engineering, vol. 34, no. 8, pp. 3549–3568, 2022.

[23] S. Pan, L. Luo, Y. Wang, C. Chen, J. Wang, and X. Wu, “Unifying large language models and knowledge graphs: A roadmap,” IEEE Transactions on Knowledge and Data Engineering, vol. 36, no. 7, pp. 3580–3599, 2024.

[24] L. Wu et al., “A survey on large language models for recommendation,” World Wide Web, vol. 27, no. 5, Art. no. 60, 2024.

[25] “Retrieval-augmented generation for knowledge-intensive language applications,” Peer-Reviewed Journal of Computer Science (PRJCS), E-ISSN 3139-5082, 2025.

[26] “Explainable artificial intelligence in critical decision-making systems,” Peer-Reviewed Journal of Computer Science (PRJCS), E-ISSN 3139-5082, 2025.