Real-time Commentator Assistant for Photo Editing Live Streaming

Matīss Rikters and Goran Topić
National Institute of Advanced Industrial Science and Technology
{firstname.lastname}@aist.go.jp

Abstract

Live commentary has the potential of making specific broadcasts such as sports or video games more engaging and interesting to watch for spectators. With the recent popularity rise of online live streaming many new categories have entered the space, like art in its many forms or even software development, however, not all live streamers have the capability to be naturally engaging with the audience. We introduce a live commentator assistant system that can discuss what is visible on screen in real time. Our experimental setting is focused on the use-case of a photo editing live stream. We compare several recent vision language models for commentary generation and text to speech models for spoken output, all on relatively modest consumer hardware configurations.

1 Introduction

Introducing live commentary to a broadcast can meaningfully impact the enjoyment of its viewers, but being a fun and engaging commentator is a skill that many people simply do not possess. Existing systems can sometimes require a significant amount of compute and often cannot function in real-time on consumer hardware. Our goal is to build a virtual commentator assistant system that would be capable of functioning on a consumer-level laptop or desktop while also performing other resource-intensive tasks in the background or foreground such as photo editing and live streaming software.

We choose the domain of photo editing live streaming for its simplicity and somewhat slow nature. This allows us to sample the screen with a far lower frequency, enabling all necessary computation to run on-device in adequate time. Aside from photo editing, there are many other slow-paced categories for live streaming, such as making miniature models, software and game development, or even cooking. Additionally, the popularity of VTubers has tremendously grown in the live streaming space, which are online entertainers who use virtual avatars instead of showing themselves directly live on camera. This inspires us to also create a visual animated character who would be personified to speak out the generated comments.

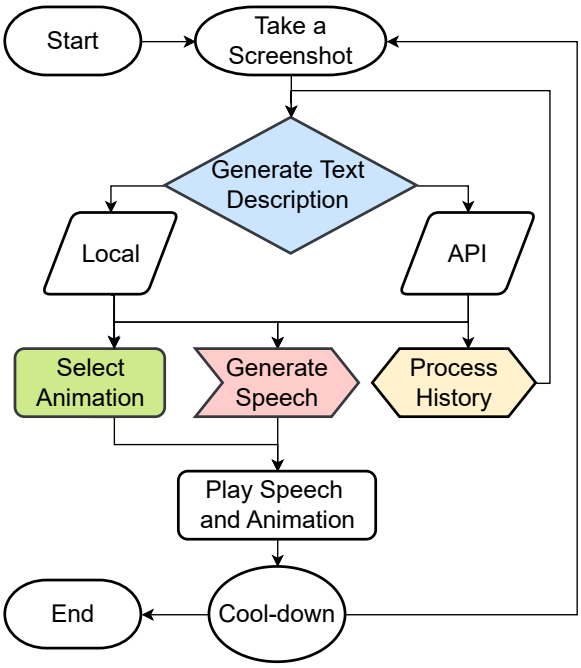

Figure 1: Live commentary system overview flowchart.

2 Related Work

Previous work (Ishigaki et al., 2023) has focused on generating audio commentary from game telemetry data, specifically – a racing game which allows such data collection. Unlike our work, their method does not consider what is actually shown on the screen. They also use a separate server for calculations and only produce audio output.

Yamazaki et al. (2023) propose an open-domain avatar chatbot in a virtual reality environment, which incorporates speech recognition, modules for spoken text processing and refinement, avatar

Proceedings of The 14th International Joint Conference on Natural Language Processing and The 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics: System Demonstrations, pages 17–24
December 20-24, 2025 ©2025 Association for Computational Linguistics

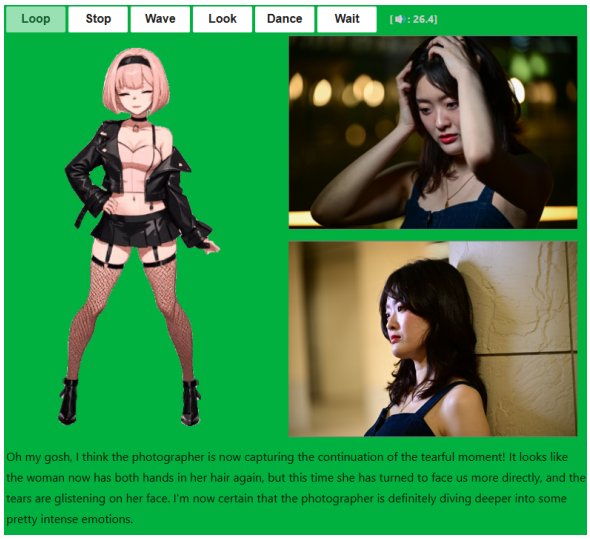

Figure 2: A screenshot of the user interface for control. The first row shows the main control buttons for character animations, a status timer showing how many seconds it takes to generate text, speech, and how long until all speech is finished playing back, and the cool-down timer, if enabled. Further down is the animated character, current and previous photo (screenshot), and the generated text comment to be spoken out.

expression generation, text to speech (TTS), as well as use of an LLM for text comprehension and generation. While the authors mention that using an 82B parameter LLM and an in-house TTS solution is challenging for generation speed, there is no detail on what hardware is used or processing time for each component.

Marrese-Taylor et al. (2022) sample suitable utterances from open-domain input videos and generate textual commentary to enhance the viewing experience. However, their approach is not real-time and the nature of open-domain videos makes the task much more difficult to tackle.

3 Architecture

The system consists of three main components as highlighted in Figure 1 - text generation from the screenshot, speech generation from the text and visual expression animation based on the text contents. After clicking the 'Loop' button (top-left in Figure 2), the process begins and keeps running until the 'Stop' button is pressed. A screenshot is automatically taken, a text description is generated based on the screenshot, and speech is generated based on the text, along with selecting an appropriate animation for the assistant character to display while the speech is being played.

After a configurable cool-down period, the process starts again; however, in subsequent iterations, the model is also supplied with the last screenshot and previously generated comments, to contextualise the current screenshot. To avoid the history becoming too large and overburdening the model, a history compaction process is also implemented, to summarise history and compact it to a predefined size. Figure 3 shows the system at work with the animated avatar overlay on top of the photo editing view, as well as the current screenshot on the bottom left and the previous screenshot on the bottom right side.

3.1 Spoken Text Generation

For the task of generating spoken text based on the screenshot of the photo currently being edited, we use a smaller-sized multimodal large language model (LLM) on device to accommodate a good balance of memory usage and relative performance. By default we choose *Phi-3.5-vision-instruct* (Abdin et al., 2024), but the model parameter is configurable with support for several other similar models, which are loaded from the Hugging Face model library[1]. Model performance is compared in the Experiments and Results sections. We also enable the use of online API versions of multimodal LLMs such as *Gemini-2.5-flash* by Google or *GPT 4o* by OpenAI to offload this more intensive task in GPU-poor scenarios.

The comments get generated from the first screenshot based on the default prompt (listed in the Appendix), and then from a combination of the most recent current screenshot and the one prior. Previous generated comments are maintained as context, and optionally summarised in a compact history representation after a certain configurable threshold.

3.2 History Compacting

By default, all the comments generated by the model are kept in the memory, and provided to the model as context for the user's current activity. If left unchecked, after a while this will result in more time spent in comment generation, as well as incur larger fees if a paid remote API is used. To control this, a history size range can be specified. If a maximum size is set, then the history will be compacted when this size (in number of comments) is reached. The most recent comments, up

[1] https://huggingface.co/models

Figure 3: A screenshot of the live-stream view with the assistant character and current/previous photos overlay.

to the minimum size setting, will be retained as-is, while the older comments will be summarised into a single comment representing old history.

3.3 Speech Generation

When searching for a viable approach for speech generation, we set a criterion that the speech generated by the model should sound more like a fictional character than an actual person, along with having capabilities to generate speech with emotion instead of being monotone. However, the main criterion was model size and generation speed, while maintaining reasonable output quality.

We found that the *Kokoro-82M* model[2] performs amazingly well for its size and also allows for some customisation of the generated voice. We compare it with several other text to speech (TTS) models in sizes up to 350M parameters in the Experiments and Results sections.

3.4 Visual Character and Animation

To generate the visual character, we used the AI Anime Generator[3] on Perchance (a platform for creating and sharing random generators) using Stable Diffusion (Rombach et al., 2022) as a backend. The character was generated based on a prompt describing a selection of unique visual features and a simple uniform background for easy removal.

Next, we used the *Wan2.1-I2V-14B-720P* model (Wang et al., 2025) to generate short animations for various situations. We generated several variations of the character talking calmly to enrich the diversity of expressions shown. We also prompted the model to generate specific animations for occasions when the character would express particular interest, happiness, fear, affection and other emotions. In addition, we prepared animation versions for the character to enter the screen from the side; leave the screen; wave hello or good bye; wait patiently for the next time to start speaking; look around to both sides pretending to be bored; and slowly dance while nothing interesting is happening.

Running specific animations can be controlled through the user interface (UI). However most are selected automatically, depending on the contents of the generated text. The selection of specific animations expressing interest, happiness, fear, affection and other emotions is based on pre-defined regular expressions. For example, the regular expression to trigger the animation showing the character being scared is "*\b(?:scar\w+|creep\w*|fright\w*|spook\w*)\b*".

[2] https://huggingface.co/hexgrad/Kokoro-82M
[3] https://perchance.org/ai-anime-generator

Model	Size	R3090	G1650L	R3070L	R4070L	R4090L	M3 Pro	Average
Phi-3.5	4.2B	9.8	111.0	11.0	13.6	10.0	103.4	43.1
Phi-4	5.6B	14.2	-	1360.6	1270.2	11.0	-	664.0
Gemma 3	4B	25.9	548.3	25.5	34.6	25.1	38.3	116.3
	12B	31.4	-	421.0	510.2	32.2	-	248.7
Qwen 2.5-VL	3B	9.9	247.2	12.4	19.3	11.1	20.7	53.4
	7B	10.2	-	12.4	32.9	10.6	-	16.5
FastVLM	0.5B	9.1	13.7	6.7	11.0	8.7	7.3	9.4
	1.5B	11.3	-	8.4	16.0	11.6	12.9	12.0
	7B	12.1	-	12.9	25.8	11.9	-	15.7
	Average	14.9	230.0	207.9	214.8	14.7	36.5	

Table 1: Results on text description generation in seconds on select consumer hardware. We abbreviate RTX and GTX to R and G, and Laptop to L for the NVIDIA GPU models. All results are averages over 30 runs. A dash represents unsuccessful runs for the specific model-hardware combination.

		1 Image	2 Images
Phi-3.5	4.2B	363.1	632.5
Phi-4	5.6B	385.4	682.7
Gemma	4B	534.8	727.1
	12B	429.8	505.1
Qwen2.5-VL	3B	458.1	582.0
	7B	395.3	615.6
FastVLM	0.5B	690.9	779.0
	1.5B	706.1	891.5
	7B	816.3	823.7
	Average	531.1	693.2

Table 2: Average text length in characters with one or two images as inputs. All averages over 30 runs.

4 Experiments

We experiment with testing the system on four consumer-grade gaming laptops with NVIDIA GTX 1650 4GB, RTX 3070 8GB, RTX 4070 8GB and RTX 4090 16GB GPUs, one desktop with RTX 3090 24GB, and a Macbook with M3 Pro and 18GB of memory. We run the experiments in two different settings - 1) running all models locally; and 2) running speech generation locally, but relying on Google Gemini[4] or OpenAI GPT[5] for text generation.

For text generation, we compare four different multimodal language model families and test model sizes from 0.5B to 12B parameters. The models are compared on generation speed, length of the generated comments, and also how much do newly generated comments overlap with previously generated comments in the same session.

Meanwhile in terms of speech models, we only consider models up to 350M parameters in size, as anything larger takes too much time and GPU memory to feasibly be part of our system. TTS models are mainly compared in terms of generation speed.

Flash attention (Dao, 2024) is used on compatible hardware (everything except M3 Pro and GTX 1650). All models are loaded with 4-bit precision (Dettmers et al., 2023) if compatible (everything except M3 Pro and the Phi-4 model).

A short demonstration video recording is available on YouTube[6]. We also release our source code in a public GitHub repository under a permissive license[7].

5 Results

The experiment results maily help us validate model compatibility with reasonable consumer hardware, as well as validate our choices of default models and other supported models. We consider the laptop with the RTX 4090 Laptop GPU as our main baseline hardware configuration, the GTX 1650 Laptop GPU – our minimum configuration, and the M3 Pro – our alternative configuration.

5.1 Text Generation

For the task of comment generation, we compare the following versions of recent multimodal LLMs – Phi 3.5 Vision Instruct (Abdin et al., 2024), Phi

[4]Gemini 2.5 Flash-Lite (August 2025) - https://ai.google.dev/gemini-api/docs/models#gemini-2.5-flash-lite

[5]GPT-4o mini (August 2025) https://platform.openai.com/docs/models/gpt-4o-mini

[6]https://www.youtube.com/watch?v=LuPcfsqPSso

[7]https://github.com/M4t1ss/live-photo-commentary

Model	Size	History 0 tok	History 0 chr	History 1 tok	History 1 chr	History 5 tok	History 5 chr	Average tok	Average chr
Phi-3.5	4.2B	53.1%	8.3%	55.8%	8.2%	80.4%	19.7%	63.1%	12.1%
Phi-4	5.6B	56.9%	7.2%	70.2%	6.5%	91.6%	53.2%	72.9%	22.3%
Gemma	4B	55.6%	12.4%	51.1%	8.3%	59.0%	9.5%	55.2%	10.1%
	12B	50.4%	11.8%	46.8%	9.5%	49.1%	9.6%	48.8%	10.3%
Qwen2.5-VL	3B	52.8%	6.9%	46.0%	7.4%	71.3%	22.5%	56.7%	12.3%
	7B	50.4%	8.1%	48.7%	7.7%	45.4%	9.0%	48.2%	**8.3%**
FastVLM	0.5B	56.0%	10.1%	93.2%	53.5%	99.3%	82.3%	82.9%	48.7%
	1.5B	59.4%	9.2%	90.1%	24.4%	98.5%	47.0%	82.7%	26.9%
	7B	57.3%	8.1%	79.0%	7.9%	87.5%	35.8%	74.6%	17.3%
Gemini	2.5-flash	43.2%	8.5%	39.8%	8.0%	43.6%	7.6%	**42.2%**	**8.0%**
GPT	4o-mini	52.1%	12.9%	51.4%	11.0%	48.1%	12.9%	50.5%	12.3%
	Average	54.7%	9.1%	64.5%	14.8%	75.8%	32.1%		

Table 3: Average text overlap between the last two generated comments with different history compacting thresholds in terms of overlapping tokens (tok) and character substring overlap (chr).

4 Multimodal Instruct (Abouelenin et al., 2025), Gemma 3 (Kamath et al., 2025) in 4B and 12B sizes, Qwen 2.5-VL (Bai et al., 2025) in 3B and 7B sizes, and FastVLM (Vasu et al., 2025) in 0.5B, 1.5B and 7B sizes.

As can be seen in table 1, all tested models run smoothly on the baseline configuration and generate comments within 10-11 seconds, aside from the two Gemma 3 models, which overall seem to be among the slowest on all tested hardware. However, both on the minimum and the alternative configurations Phi 4, Gemma 3 12B, Qwen 2.5-VL 7B and FastVLM 7B are entirely unable to run. Out of all models tested, Phi 4 has the least compatibility – unable to run on two GPUs and on two others taking over 20 minutes to produce a result.

Table 2 shows that the FastVLM models tend to generate longer outputs regardless whether the input is one image or two. The other models generate comments with an average length of 446 characters while the average for FastVLM models is 686 characters. Comparing two image inputs to one image input, the increase in average generated characters is around 160, with outliers like Phi 3.5 almost doubling the amount compared to single image, and Gemma 3 12B only generating around 40 characters more for dual image inputs.

Upon manual inspection of the generated comments, we noticed that at times there is substantial overlap between the current and previous comments generated by some models, or even a 1:1 copy – not based on the input images at all. Therefore, we evaluated handling of our history feature by the LLMs using a set of 10 consecutive photos from the same photo shoot as inputs. In table 3 we look at how the multimodal LLMs handle our history compacting approach, by comparing passing history lengths of 0 (history turned off), 1, and 5 previously generated comments along with the prompt. We measure the percentage of overlapping tokens (words) between two consecutive outputs, as well as the percentage of overleaping characters. The FastVLM models are overall worst in terms ov both overlap metrics, while Gemma 3 and Qwen 2.5-VL models along with Gemini and GPT APIs perform best here. Some models like both Phi models, Qwen 2.5-VL 3B, and FastVLM 7B only suffer from the overlapping output issue with the longer history of 5 comments.

Model	Size	Time, s
Bark	300M	48.8
Bark-small	80M	27.5
Kokoro-82M	82M	**0.5**
OuteTTS-0.1	350M	87.6
MMS-TTS-ENG	36M	5.8
SpeechT5	145M	7.5

Table 4: Comparison of several smaller text to speech models, showing model size and average generation time over 10 runs. Input text was on average 555 characters long ranging between 347 and 828 characters.

5.2 Speech Generation

Most modern TTS models have at least 0.5B parameters, which can hinder efficient execution on consumer hardware. We test base and small ver-

sions of the Bark model from Suno[8], Kokoro-82M, OuteTTs-0.1[9], MMS-TTS-ENG (Pratap et al., 2024), and SpeechT5 (Ao et al., 2022). As these models are quite small in terms of parameters, we only consider testing on our baseline hardware configuration. All tests were performed on 10 previously generated comments from Phi 4, ranging between 347 and 828 characters in length.

Table 4 shows that the Kokoro-82M is by far the overall fastest TTS model, taking on average only 0.5 seconds to generate speech for the previously generated comments from the multimodal LLMs, which is over 10 times faster than the next fastest – MMS-TTS. Based on these results, we select Kokoro-82M as the default model and add MMS-TTS-ENG and SpeechT5 as alternative options to select in our system.

6 Conclusion

In this paper, we introduce a live commentator assistant system for the use case of photo editing online live streaming, which generates real-time commentary based on what is visible on-screen. It is capable of fully functioning locally alongside live streaming and photo editing software with moderate consumer hardware requirements, as well as utilising multimodal LLM APIs to offload a major part of the required computation supporting even lower-grade hardware.

For future work we consider a wide range of potential improvements and expansions of our proposed task. In terms of expanding the scope of the task, we plan on utilising dialogue-styled commentary based on either user input or live-stream chat to make the interaction even more engaging. Other avenues of expanding the scope include enhancing live streams with additional explanatory graphics as on-screen overlays, and exploring the applicability of short video capture commentary. As for the actual quality of the generated text, we plan on performing a small-scaled human evaluation study to obtain a broader insight on the quality of the generated text beyond token and character overlap. Further improvements of output quality may also be achieved by implementing output filtering based on heuristics (Rikters, 2018) or the model attention mechanism output (Rikters and Fishel, 2017).

Limitations

In this work, we only considered using models that are publicly available at no cost to enable reproducibility. The computation setup for our experiments is relatively modest and well within reach for most who would be willing to reproduce our experiments.

Our proposed system is easily reproducible with publicly available model checkpoints and open-source tools which are cited in this paper. Our full workflow shall be released on GitHub. For the blind submission, we prepared an anonymised version as an attachment. Our system is also not limited to the specific model families cited in the paper, so one could simply swap out the text or speech models for others compatible with the Hugging Face Transformers workflow.

Ethical Considerations

Our work is fully in accordance with the ACL Code of Ethics[10]. We use only publicly available open-weight models and relatively low compute amounts while conducting our experiments to enable reproducibility. We do not conduct studies on other humans or animals in this research.

References

Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, Alon Benhaim, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Martin Cai, Qin Cai, Vishrav Chaudhary, Dong Chen, Dongdong Chen, and 110 others. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *Preprint*, arXiv:2404.14219.

Abdelrahman Abouelenin, Atabak Ashfaq, Adam Atkinson, Hany Awadalla, Nguyen Bach, Jianmin Bao, Alon Benhaim, Martin Cai, Vishrav Chaudhary, Congcong Chen, Dong Chen, Dongdong Chen, Junkun Chen, Weizhu Chen, Yen-Chun Chen, Yi ling Chen, Qi Dai, Xiyang Dai, Ruchao Fan, and 55 others. 2025. Phi-4-mini technical report: Compact yet powerful multimodal language models via mixture-of-loras. *Preprint*, arXiv:2503.01743.

Junyi Ao, Rui Wang, Long Zhou, Chengyi Wang, Shuo Ren, Yu Wu, Shujie Liu, Tom Ko, Qing Li, Yu Zhang, Zhihua Wei, Yao Qian, Jinyu Li, and Furu Wei. 2022. SpeechT5: Unified-modal encoder-decoder

[8] https://github.com/suno-ai/bark

[9] https://github.com/edwko/OuteTTS

[10] https://www.aclweb.org/portal/content/acl-code-ethics

pre-training for spoken language processing. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5723–5738, Dublin, Ireland. Association for Computational Linguistics.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, and 8 others. 2025. Qwen2.5-vl technical report. *Preprint*, arXiv:2502.13923.

Tri Dao. 2024. FlashAttention-2: Faster attention with better parallelism and work partitioning. In *International Conference on Learning Representations (ICLR)*.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. In *Advances in Neural Information Processing Systems*, volume 36, pages 10088–10115. Curran Associates, Inc.

Tatsuya Ishigaki, Goran Topić, Yumi Hamazono, Ichiro Kobayashi, Yusuke Miyao, and Hiroya Takamura. 2023. Audio commentary system for real-time racing game play. In *Proceedings of the 16th International Natural Language Generation Conference: System Demonstrations*, pages 9–10, Prague, Czechia. Association for Computational Linguistics.

Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, Gaël Liu, and 196 others. 2025. Gemma 3 technical report. *Preprint*, arXiv:2503.19786.

Edison Marrese-Taylor, Yumi Hamazono, Tatsuya Ishigaki, Goran Topić, Yusuke Miyao, Ichiro Kobayashi, and Hiroya Takamura. 2022. Open-domain video commentary generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 7326–7339, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Vineel Pratap, Andros Tjandra, Bowen Shi, Paden Tomasello, Arun Babu, Sayani Kundu, Ali Elkahky, Zhaoheng Ni, Apoorv Vyas, Maryam Fazel-Zarandi, Alexei Baevski, Yossi Adi, Xiaohui Zhang, Wei-Ning Hsu, Alexis Conneau, and Michael Auli. 2024. Scaling speech technology to 1,000+ languages. *J. Mach. Learn. Res.*, 25(1).

Matīss Rikters. 2018. Impact of Corpora Quality on Neural Machine Translation. In *Proceedings of the 8th Conference Human Language Technologies - The Baltic Perspective (Baltic HLT 2018)*, Tartu, Estonia.

Matīss Rikters and Mark Fishel. 2017. Confidence Through Attention. In *Proceedings of the 16th Machine Translation Summit (MT Summit 2017)*, Nagoya, Japan.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10684–10695.

Pavan Kumar Anasosalu Vasu, Fartash Faghri, Chun-Liang Li, Cem Koc, Nate True, Albert Antony, Gokul Santhanam, James Gabriel, Peter Grasch, Oncel Tuzel, and Hadi Pouransari. 2025. Fastvlm: Efficient vision encoding for vision language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, Jiayu Wang, Jingfeng Zhang, Jingren Zhou, Jinkai Wang, Jixuan Chen, Kai Zhu, Kang Zhao, Keyu Yan, Lianghua Huang, and 42 others. 2025. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*.

Takato Yamazaki, Tomoya Mizumoto, Katsumasa Yoshikawa, Masaya Ohagi, Toshiki Kawamoto, and Toshinori Sato. 2023. An open-domain avatar chatbot by exploiting a large language model. In *Proceedings of the 24th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 428–432, Prague, Czechia. Association for Computational Linguistics.

A Prompt Format

Table 5 shows examples of prompts used for generating comments. We use the exact same prompts for all models without fine-tuning them to each model individually. The default example prompts are formed in a way that works best with our proposed use-case of editing photography (e.g. mentioning gridlines, sliders, addressing the photographer). For other use-cases these prompts can be updated as needed.

Prompt	Text	Addition
SYSTEM	You are a friendly chatty commentator who likes to casually describe work done by a photographer in various details, even by pondering the implications on work, or leisure, being performed, etc. Write your response in a very personal way using personal pronouns and explaining what you see, perhaps also adding how it makes you feel. Do your best to not be repetitive in your choice of words. You MUST keep the response length to no more than three sentences. You MUST NOT mention any specific layout elements or tools that may be visible on the screen, such as gridlines or sliders.	
MAIN	Summarize what is visible in the current photo, <|image_1|>. How is it different from the previous photo, <|image_2|>? There may be some subtle differences as well. Do not describe the previous photo; assume you have described it already. It is only there for context, so you can notice the new things in the current photo. Do not mention photos explicitly; use words like 'I can see...' or 'The photographer is now...' and similar. Use the comment history for context and continuity, but the utmost priority should be on describing the current activity, as reflected in the current photo. DO NOT repeat comments from the history.	+ ENDING
FIRST	Summarize what is visible in this image: <|image_1|>	+ ENDING
ENDING	Do not at all mention any specific layout elements or tools that may be visible on the screen, such as overlays, gridlines or sliders. To adjust intonation, please add dedicated punctuation like ; : , . ! ? … ( ) " " For example, to emphasize a word or a phrase, surround it with "quotation marks". However, since the text will undergo speech synthesis, do not use anything unpronounceable, like emojis.	
COMPACT	Summarize in one short paragraph your (the assistant's) comments so far on the current activity; i.e. compact it into a single comment of comparable size to one individual original comment, that encapsulates the essence of the current activity's past. If some older comments pertain to a different activity, you can ignore them; focus only on the current activity. This is what you (the assistant) commented before:	
HISTORY	This is what you (the assistant) commented before:	

Table 5: Examples of prompts used for comment text generation. The FIRST prompt is used only at the beginning when there is just one screenshot, after which the MAIN prompt is used. The HISTORY prompt is used to maintain recent context, and COMPACT – for consolidating older history into a short summary.