Vision-Language Models on the Edge

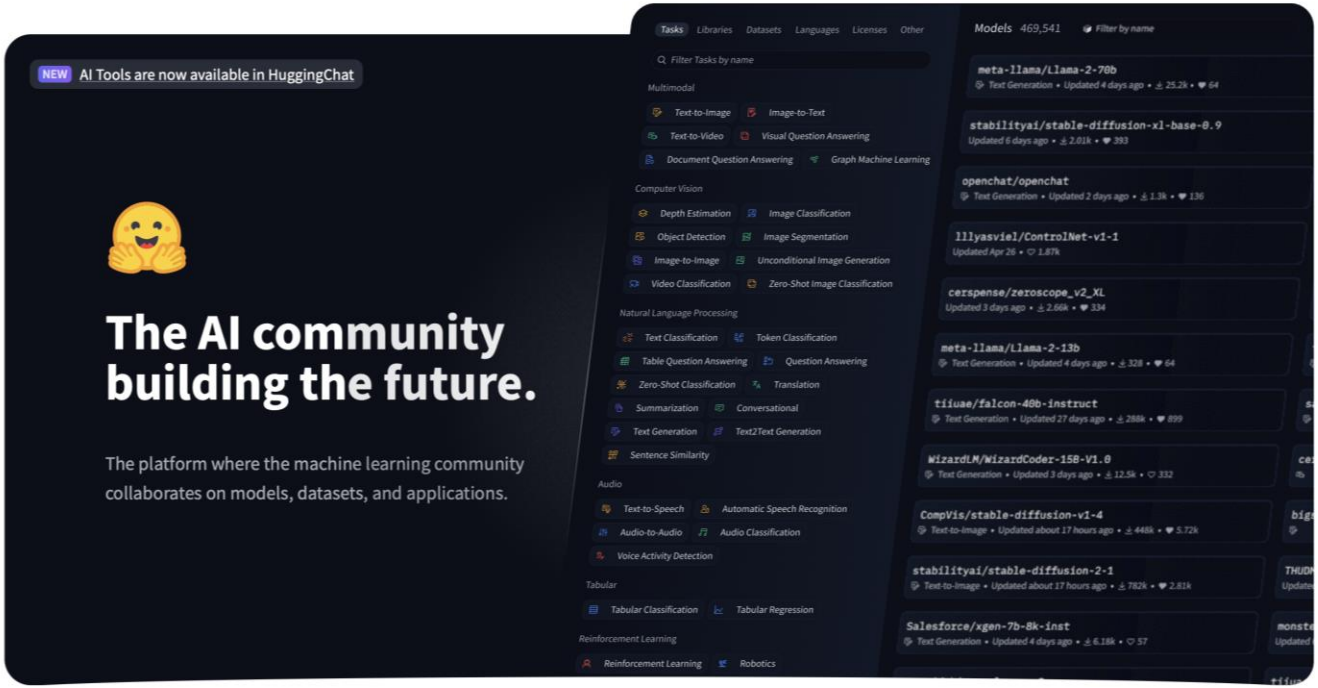
Cyril Zakka, MD
Health Lead
Hugging Face

Hugging Face

Overview

- What is Hugging Face?
- What is a VLM?
- Building SmolVLM
  - Data Preparation
  - Training
  - Evaluation and Benchmarks
- Demo

What is Hugging Face?



https://huggingface.co/

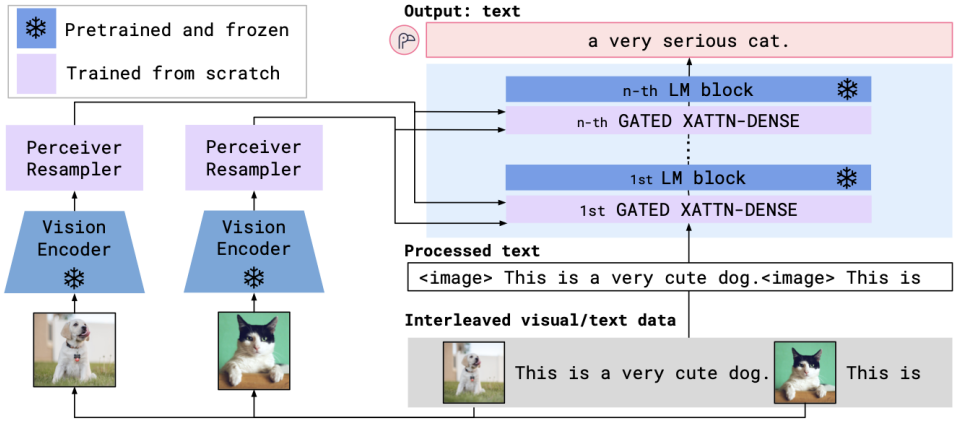
Hugging Face

What is a VLM?

- Vision Language Models (VLMs) are a type of neural network architecture trained on text and image tokens
- Text + Image + Video In → Text Out
- Two general approaches for unifying text and image tokens:
  - **Cross-Attention Architecture**
  - **Self-Attention Architecture**

What is a VLM?

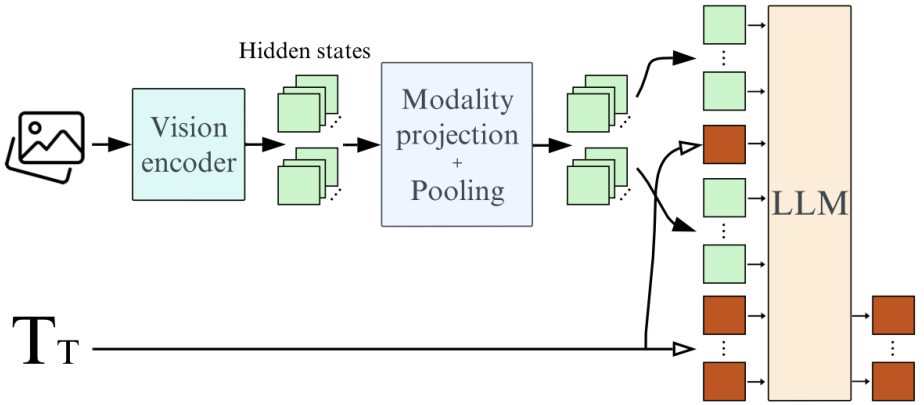
- **Cross-Attention Architecture:**
  - Frozen LM conditioned by image features from frozen vision encoder
  - Preserves LM capabilities on textual tasks
  - Only new layers are trained
  - Fine-tuning each backbone can lead to worse performance



https://arxiv.org/abs/2204.14198

What is a VLM?

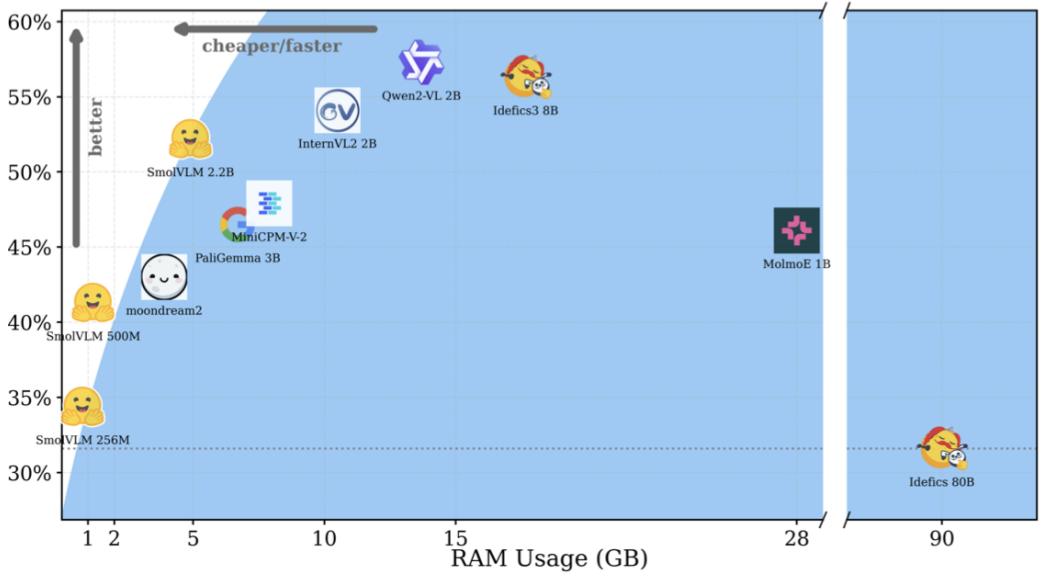
- **Self-Attention Architecture**
  - Text and image tokens in the same latent space
  - Outperforms cross-attention when trained from scratch
  - Better approach with enough data and compute
  - Most commonly adopted in SoTA models



https://arxiv.org/abs/2408.12637

Building SmolVLM

- Family of small VLMs – 256M, 500M and 2.2B parameter models
- Outperform all existing models per memory consumption
- Matches or exceeds performance of other models
- Fully open source
  - Weights + data + training code
- Available in Python and Swift



Hugging Face

SmolVLM: Data Preparation – Text

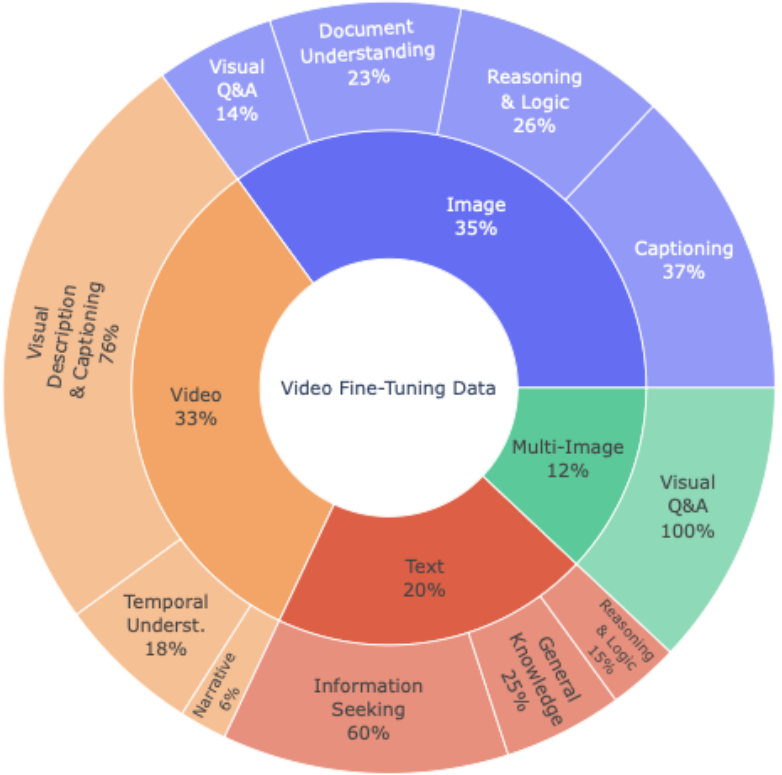
- Get large scale data – make your own or use the Common Crawl as a starting point.
- Filter data:
  - URL filtering: Blocklists, words, language classifier, repetition filters
  - Content filtering: Language, Gopher, minHash, C4 and other custom heuristics or filters
  - Other: Personally Identifiable Information (PII) removal

SmolVLM: Data Preparation – Vision Stage

- **Vision Training Data**
  - Mix of common vision tasks
  - OCR, captioning, chart understanding, table understanding etc.
  - General knowledge QA text data added to maintain performance on text

SmolVLM: Data Preparation – Video Stage

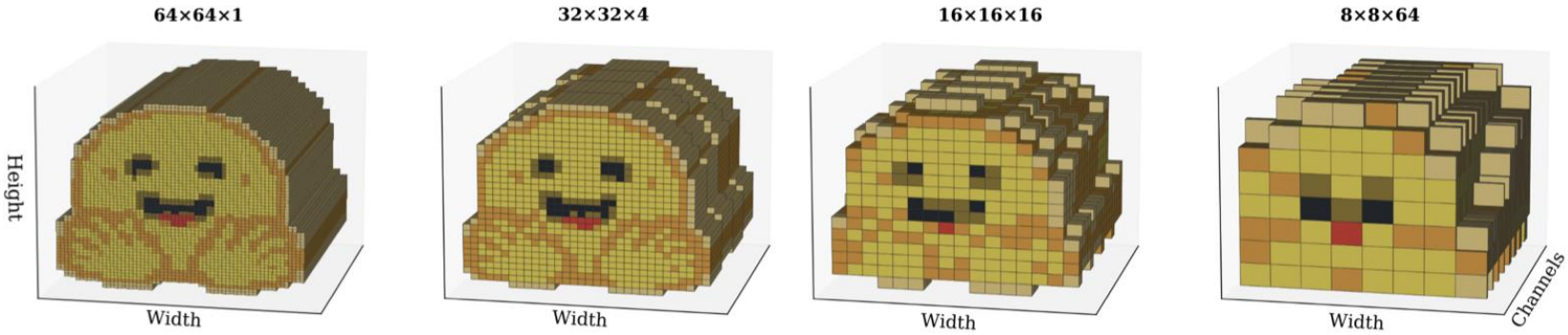
- **Video Fine-tuning Data**
  - Mix of Image (35%), Text (20%), Video (33%) and multi-image (20%).
  - Percentages determined via ablations
  - Mix of modalities to prevent catastrophic forgetting



SmolVLM: Training

- Self-attention architecture
- Balanced parameters w/ vision encoder + language model
- Images + video tokens concatenated to text tokens
  - 2K context length (CL) of LM model insufficient.
    - Increased to 16K using Rotary Positional Embedding (RoPE) + trained on long-context data (base = 273K)
    - Token compression (pixel shuffle) used to fit more tokens into CL (r=2). Can harm performance of granular tasks at higher ratios.
    - Image splitting + video frame resizing

SmolVLM: Pixel Shuffle



https://arxiv.org/pdf/2504.05299

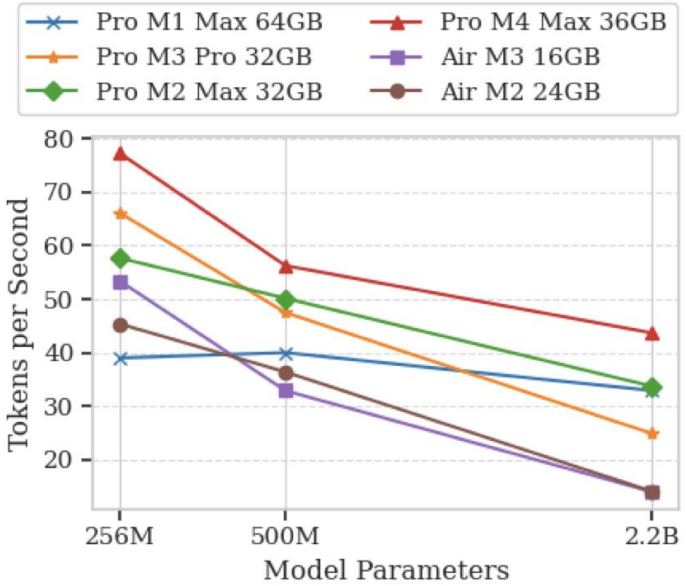
Hugging Face

SmolVLM: Family of Models

- **SmolVLM-256M:**
  - < 1 GB of GRAM
  - 93 M SigLIP-B/16 vision encoder + SmolLM2-135M LM
- **SmolVLM-500M:**
  - Balance between memory efficiency and performance
  - 93 M SigLIP-B/16 vision encoder + SmolLM2-360M LM
- **SmolVLM-2.2B:**
  - 400 M SigLIP-SO vision encoder + SmolLM2-1.7B LM
  - Maximizes performance + deployable on higher end edge devices

SmolVLM: Evaluation and Benchmarking

- Evaluated on a panel of 30 benchmarks from the literature
- The best benchmarks are the ones that reflect YOUR use case. Other benchmarks should just be used as a “sniff test”
- Biggest impact on downstream task performance is **data selection** and **data distribution**
  - Limited model capacity means optimizing for these factors
  - Determined through intuition and ablations



SmolVLM: Evaluation and Benchmarking

Capability	Benchmark	SmolVLM 256M	SmolVLM 500M	SmolVLM 2.2B	Efficient OS
Single-Image	OCRBench (Liu et al., 2024e) Character Recognition	52.6%	61.0%	72.9%	54.7% MolmoE-A1B-7B
	AI2D (Kembhavi et al., 2016) Science Diagrams	46.4%	59.2%	70.0%	71.0% MolmoE-A1B-7B
	ChartQA (Masry et al., 2022) Chart Understanding	55.6%	62.8%	68.7%	48.0% MolmoE-A1B-7B
	TextVQA (Singh et al., 2019) Text Understanding	50.2%	60.2%	73.0%	61.5% MolmoE-A1B-7B
	DocVQA (Mathew et al., 2021) Document Understanding	58.3%	70.5%	80.0%	77.7% MolmoE-A1B-7B
	ScienceQA (Lu et al., 2022) High-school Science	73.8%	80.0%	89.6%	87.5% MolmoE-A1B-7B
Multi-task	MMMU (Yue et al., 2024a) College-level Multidiscipline	29.0%	33.7%	42.0%	33.9% MolmoE-A1B-7B
	MathVista (Lu et al., 2024b) General Math Understanding	35.9%	40.1%	51.5%	37.6% MolmoE-A1B-7B
	MMStar (Chen et al., 2024a) Multidisciplinary Reasoning	34.6%	38.3%	46.0%	43.1% MolmoE-A1B-7B
Video	Video-MME (Fu et al., 2024) General Video Understanding	33.7%	42.2%	52.1%	45.0% InternVL2-2B
	MLVU (Zhou et al., 2024) MovieQA + MSRVTT-Cap	40.6%	47.3%	55.2%	48.2% InternVL2-2B
	MVBench (Li et al., 2024b) Multiview Reasoning	32.7%	39.7%	46.3%	60.2% InternVL2-2B
	WorldSense (Hong et al., 2025) Temporal + Physics	29.7%	30.6%	36.2%	32.4% Qwen2VL-7B
	TempCompass (Liu et al., 2024d) Temporal Understanding	43.1%	49.0%	53.7%	53.4% InternVL2-2B
Average	Across Benchmarks	44.0%	51.0%	59.8%	–
RAM Usage	Batch size = 1	0.8 GB	1.2 GB	4.9 GB	27.7 GB MolmoE-A1B-7B
	batch size = 64	15.0 GB	16.0 GB	49.9 GB	–

Hugging Face

SmolVLM: Demo

Conclusion

- **Data, data, data:**
  - Good data reflective of your downstream tasks goes a long way
- **Small decisions impact small models in a big way:**
  - For every change that you make to the model or training parameter, perform ablations
- **Goodhart's Law:**
  - When a measure becomes a target, it ceases to be a good measure
  - Test different models on a dataset reflective of your task

Resources

Learn More

Building and better understanding vision-language models: insights and future directions
https://arxiv.org/pdf/2408.12637

SmolVLM: Redefining small and efficient multimodal models
https://arxiv.org/pdf/2504.05299

Dataset Curation Best Practices: Fine-Web
https://huggingface.co/spaces/HuggingFaceFW/blogpost-fineweb-v1

HuggingSnap iOS Code + Demo
https://github.com/huggingface/HuggingSnap