ROB 498/599: Deep Learning for Robot Perception (DeepRob)

Lecture 22: Visualizing Models;

Vision Language Models; VLA Models

04/07/2025

https://deeprob.org/w25/

M | ROBOTICS

Today

- Feedback and Recap (5min)
- Visualizing Models (30min)
- VLM/VLAM (30min)
- Final Project logistics (5min)
- Summary and Takeaways (5min)

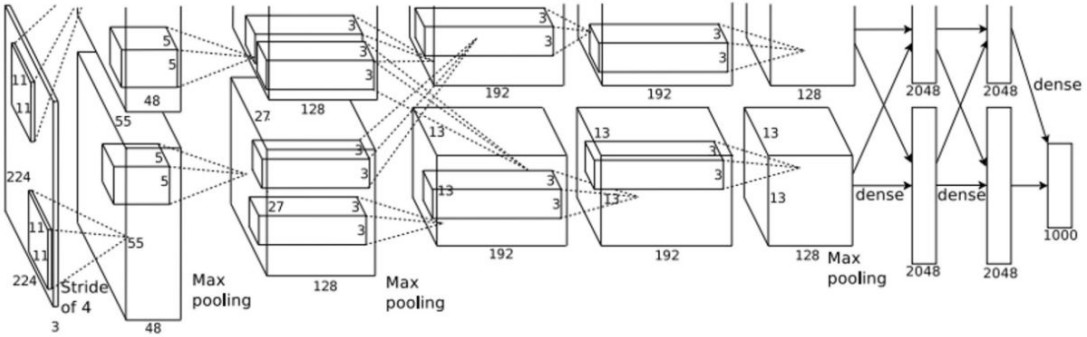
Visualizing Models

Visualizing features, networks, etc…

What's going on inside Convolutional Networks?

This image is CC0 public domain

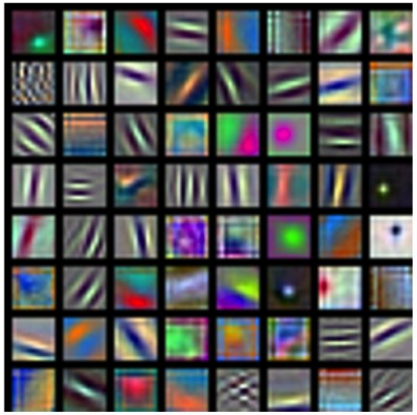
Input Image:
3 x 224 x 224



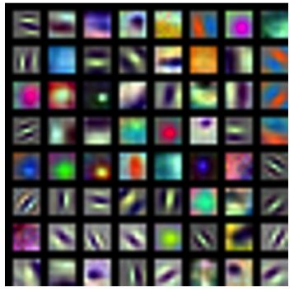
Class Scores:
1000 numbers

What are the intermediate features looking for?

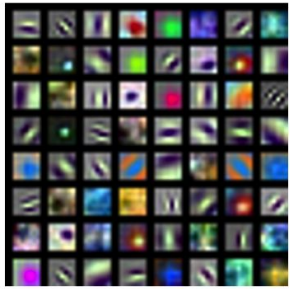
First Layer: Visualize Filters



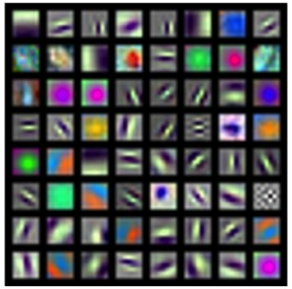
AlexNet:
64 x 3 x 11 x 11



ResNet-18:
64 x 3 x 7 x 7



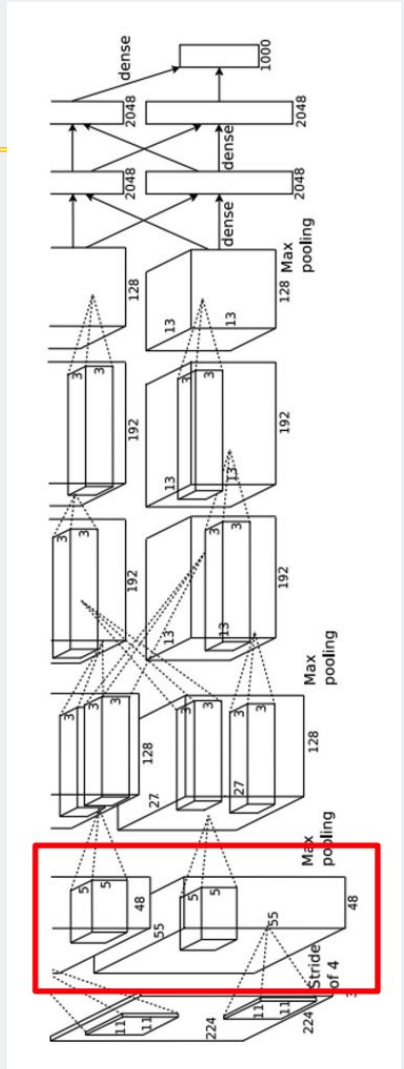
ResNet-101:
64 x 3 x 7 x 7



DenseNet-121:
64 x 3 x 7 x 7

Oriented edges
Opposing colors

https://arxiv.org/pdf/1404.5997
https://arxiv.org/pdf/1512.03385
https://openaccess.thecvf.com/content_cvpr_2017/papers/Huang_Densely_Connected_Convolutional_CVPR_2017_paper.pdf



Higher Layers: Visualize Filters

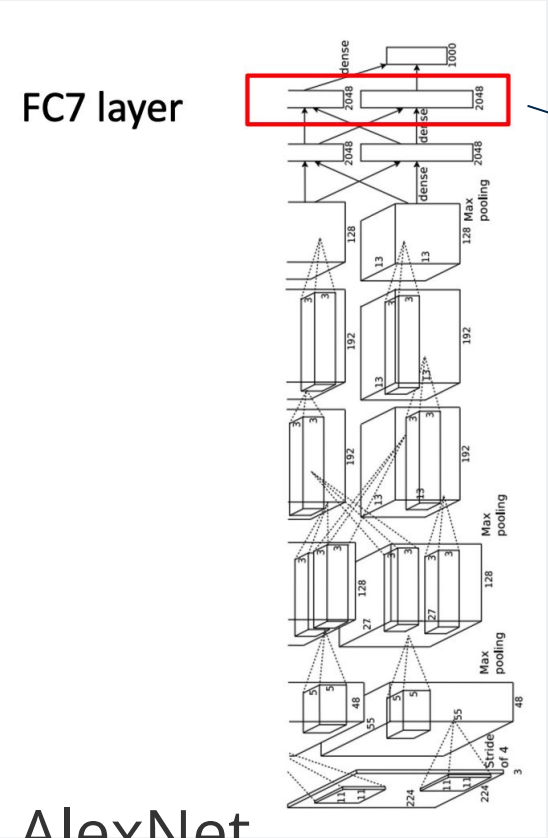
First layer weights: 16 x 3 x 7 x 7

Second layer weights:
20 x 16 x 7 x 7

Third layer weights:
20 x 20 x 7 x 7

https://cs.stanford.edu/people/karpathy/convnetjs/demo/cifar10.html

Last Layer

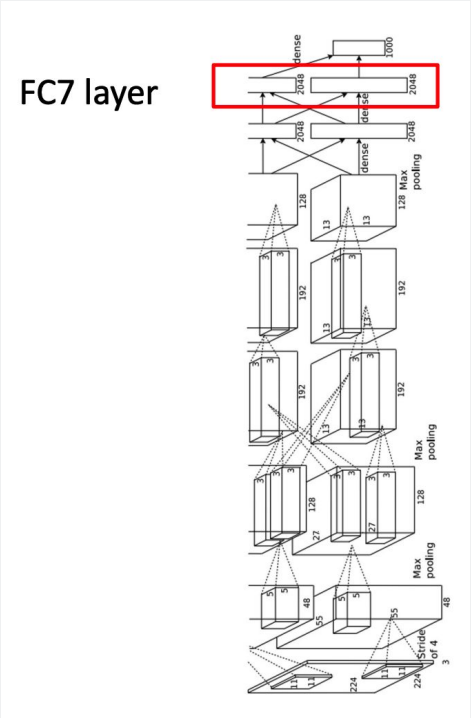


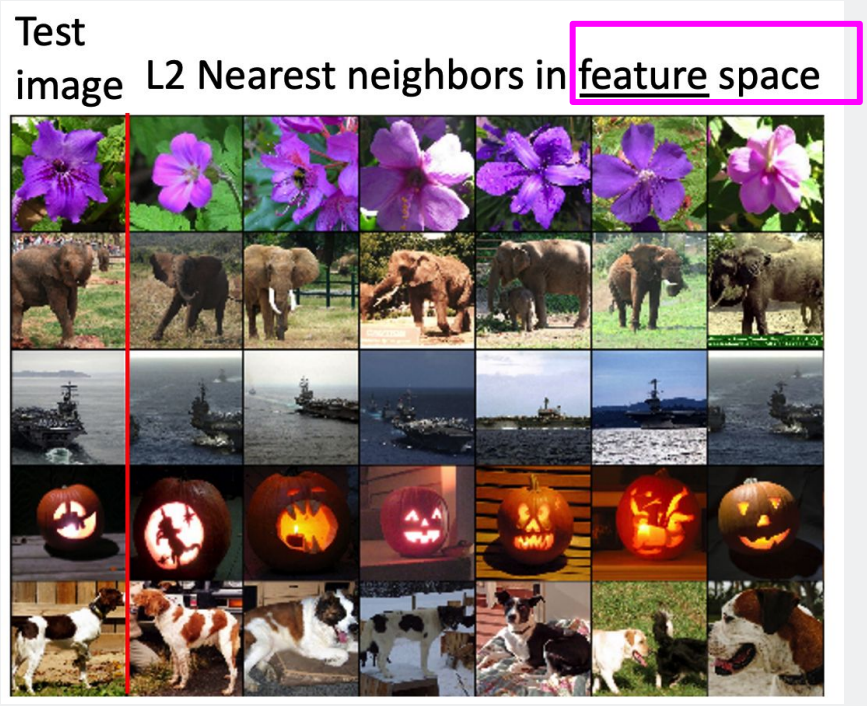
4096-dimensional feature vector for an image (layer immediately before the classifier)

Run the network on many images, collect the feature vectors

A few things we can do on the final layers

1. Nearest Neighbor Information Retrieval





FC7 feature vector space

Semantic similar

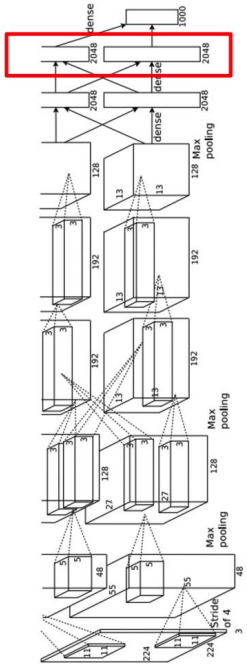
M | ROBOTICS

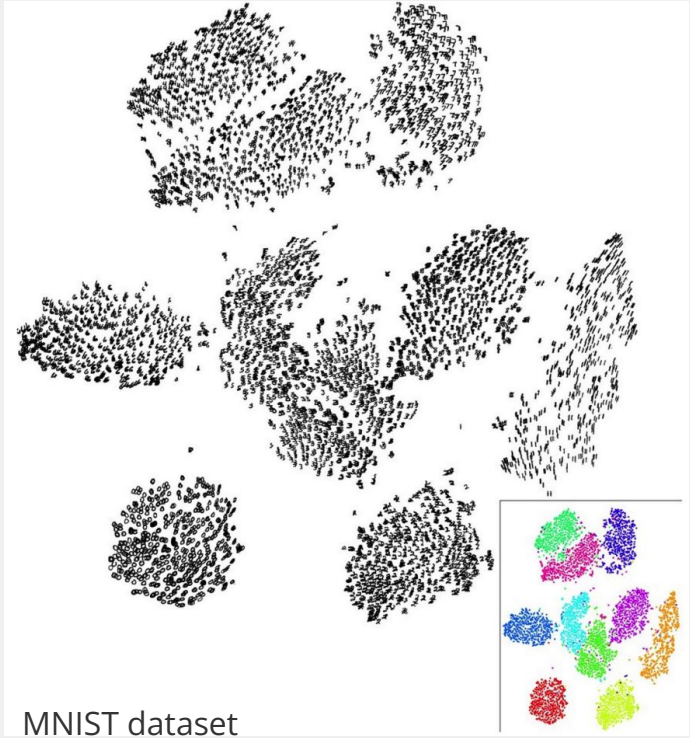
A few things we can do on the final layers

https://cs.stanford.edu/people/karpathy/cnnembed/

2. Dimensionality Reduction

FC7 layer





MNIST dataset

Visualize the “space” of FC7 feature vectors by reducing dimensionality of vectors from 4096 ⇨ 2 dimensions

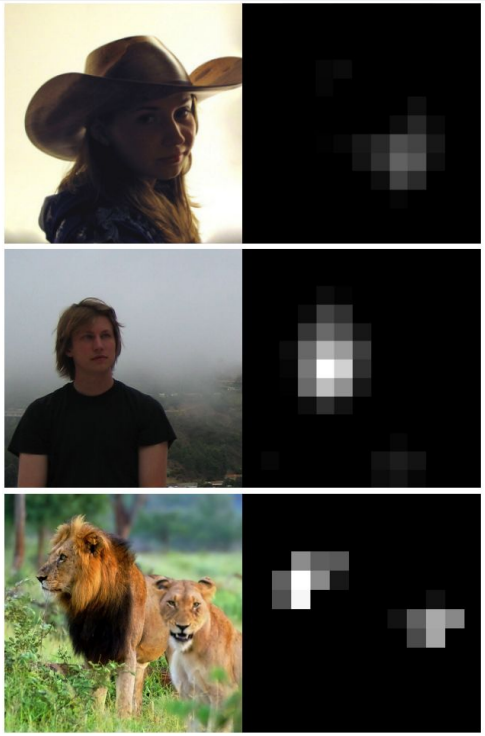
Simple algorithm: Principal Component Analysis (PCA)

More complex: t-SNE

M | ROBOTICS

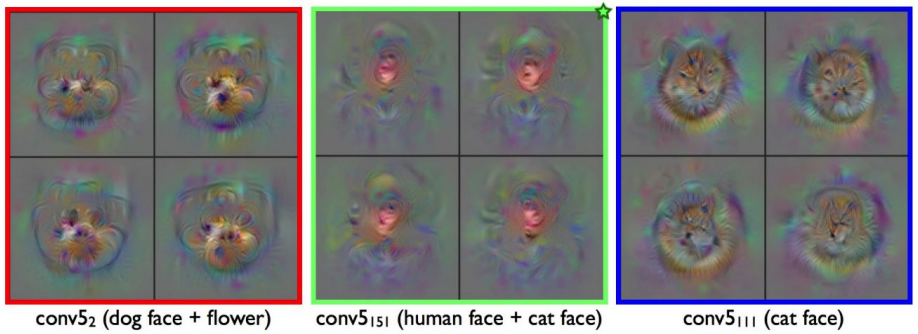
Visualizing Activations

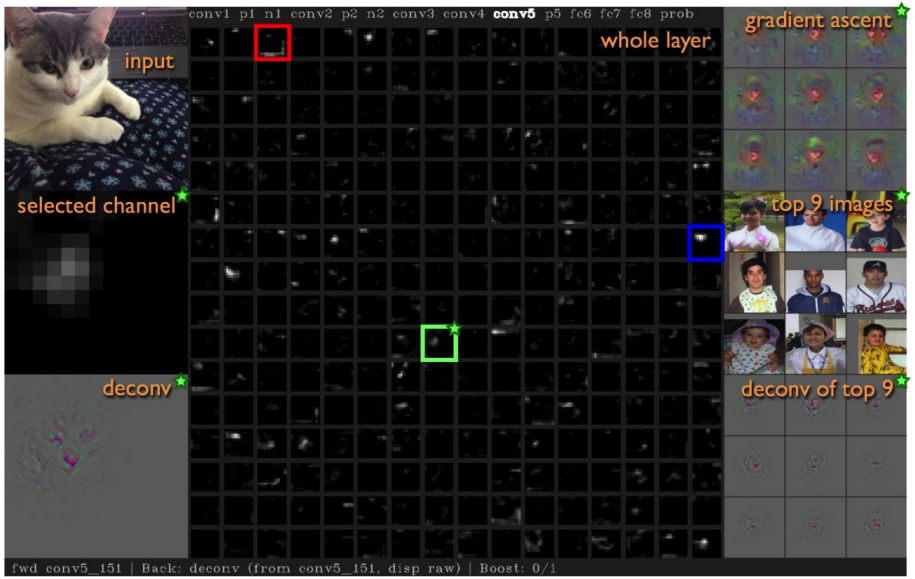
https://yosinski.com/deepvis
https://github.com/yosinski/deep-visualization-toolbox



Right: 13x13 conv5151 channel activations.

Left: 13×13 activations of the 151st channel on the conv5 layer





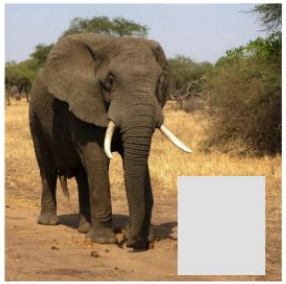
https://arxiv.org/pdf/1506.06579

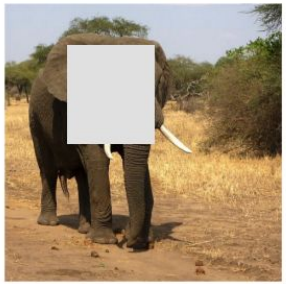
Masking

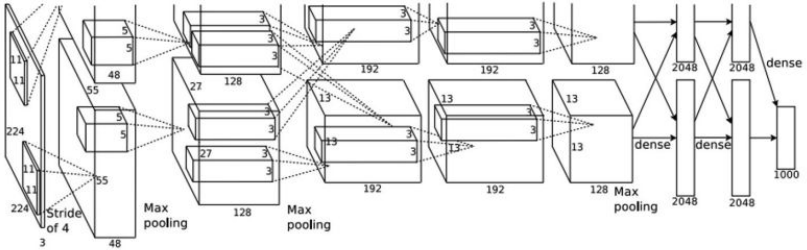
spurious correlation - be aware!

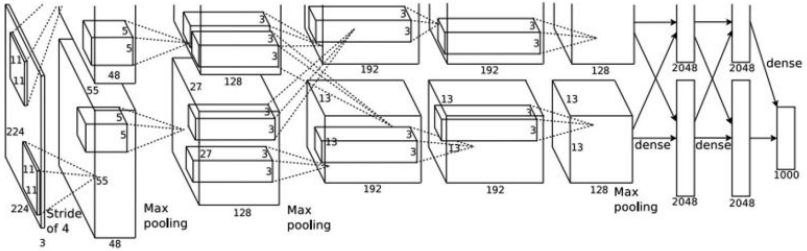
Saliency map

Mask part of the image before feeding to CNN, check how much predicted probabilities change



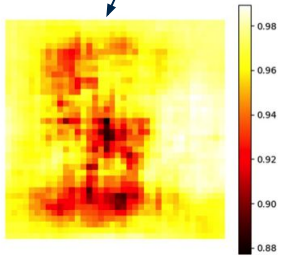




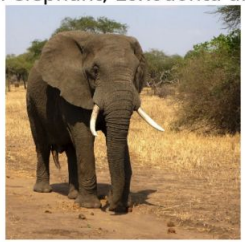


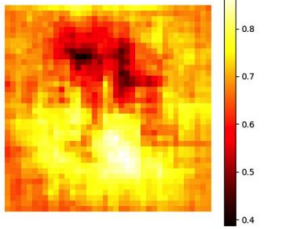
Color: predicted classification score

schooner

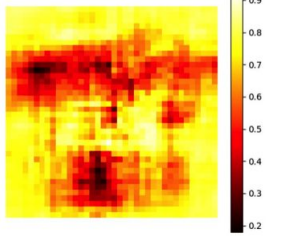


African elephant, Loxodonta africana





go-kart

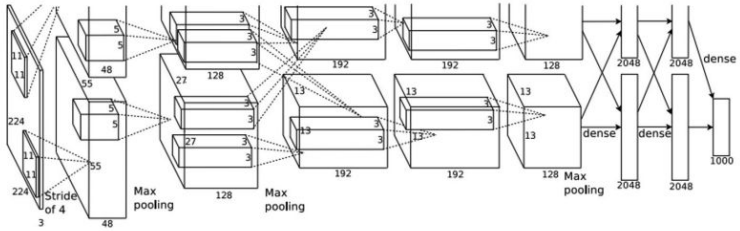


https://arxiv.org/pdf/1311.2901

M | ROBOTICS

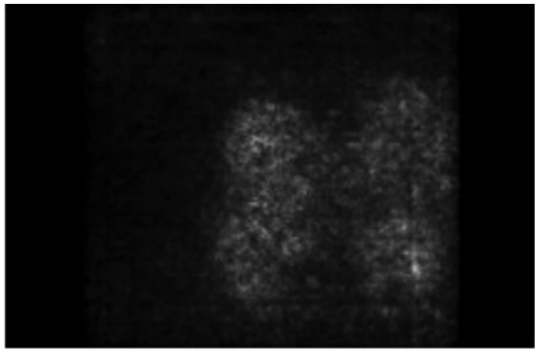
Saliency via Backprop

Forward pass: Compute probabilities



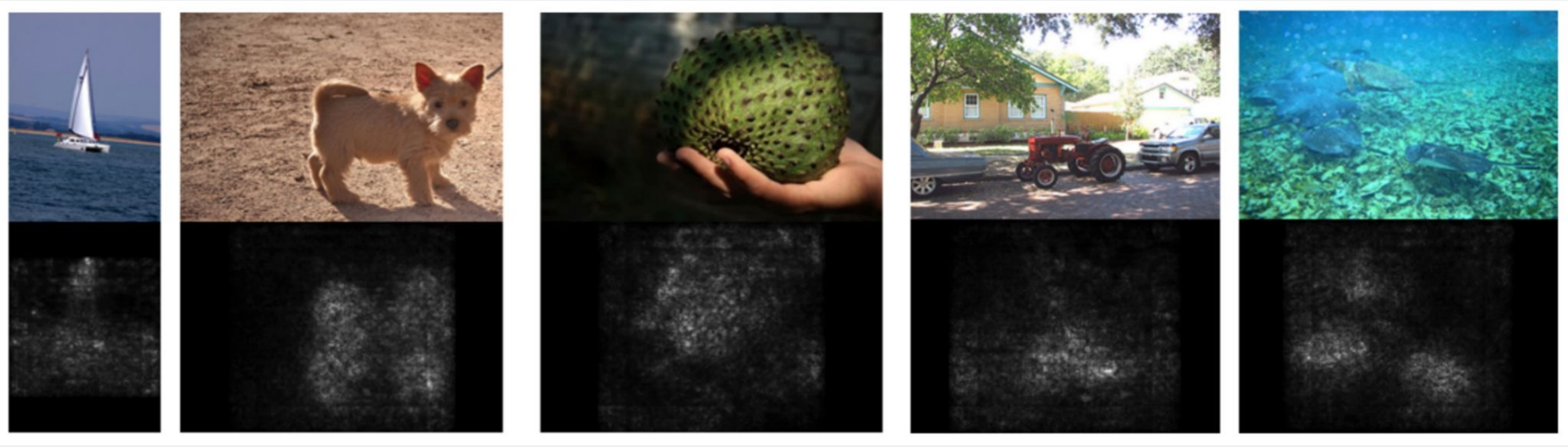
Dog

Compute gradient of (unnormalized) class score with respect to image pixels, take absolute value and max over RGB channels



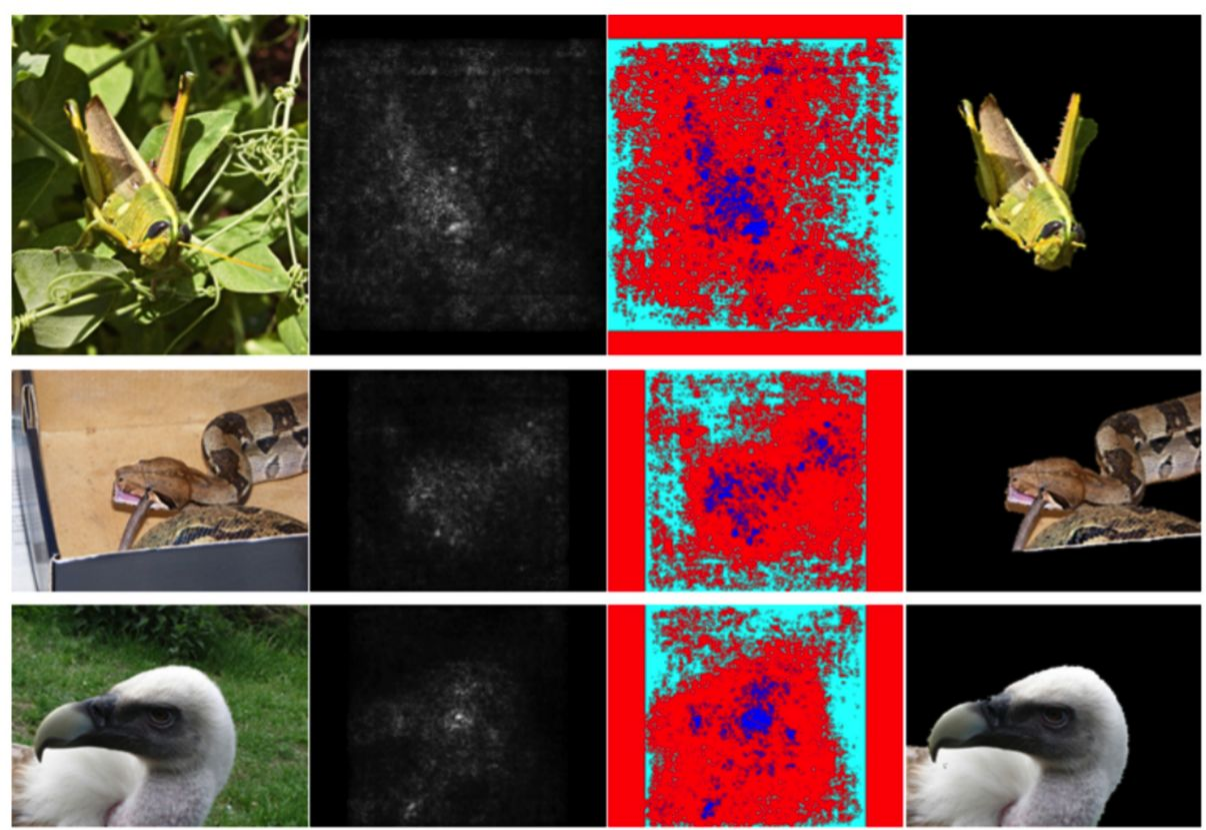
https://arxiv.org/pdf/1312.6034

Saliency via Backprop: a few more examples



ROBOTICS

Saliency Maps: Segmentation without Supervision

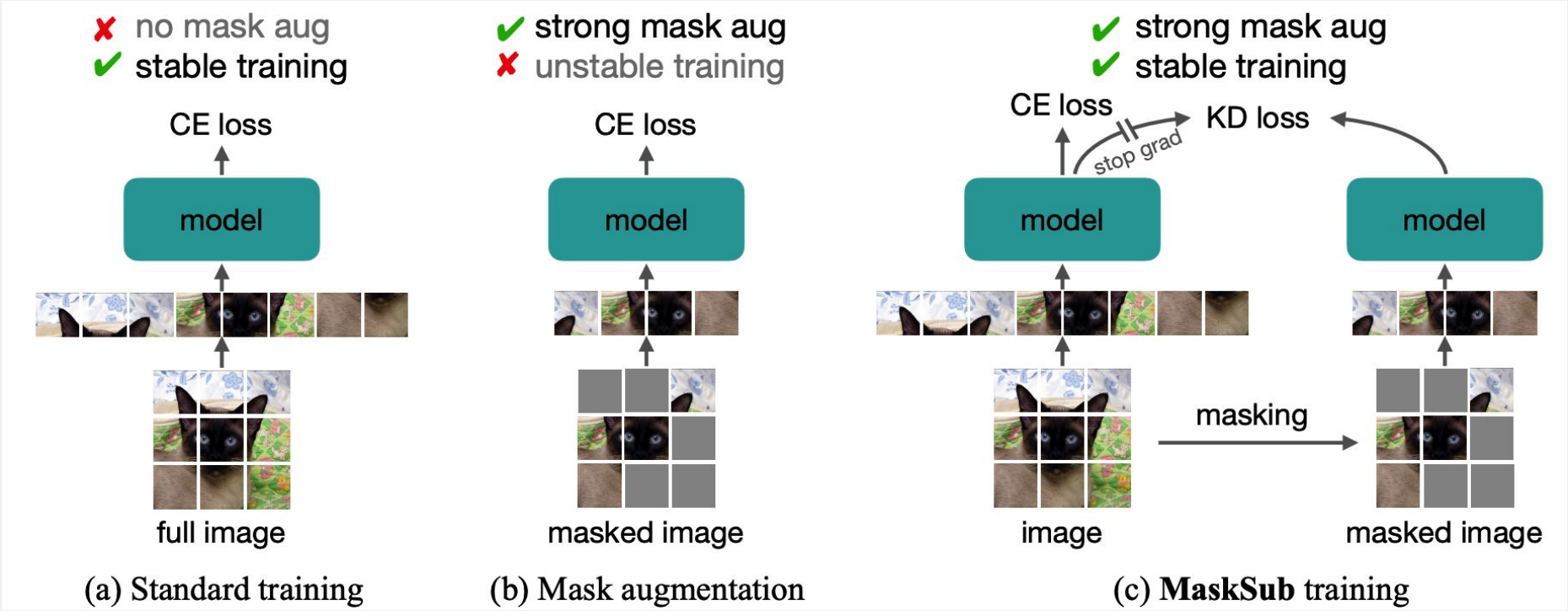


Use **GrabCut** on saliency map

https://docs.opencv.org/3.4/d8/d83/tutorial_py_grabcut.html

M | ROBOTICS

Masking Meets Supervision



https://arxiv.org/pdf/2306.11339 (CVPR 2025)

M | ROBOTICS

Masking Meets Supervision

self-distillation

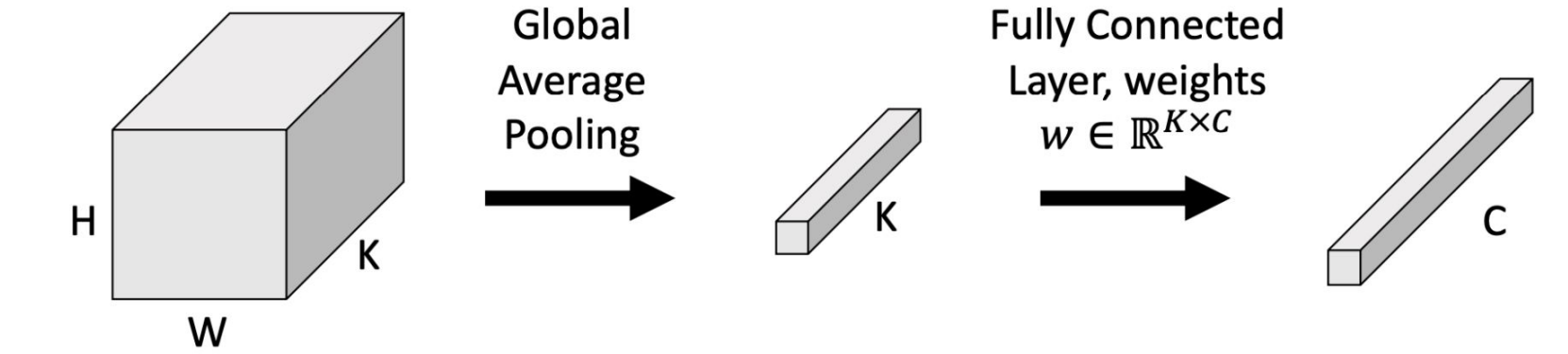
Algorithm 1 MaskSub in PyTorch-style pseudo-code

```
for (x, label) in data_loader:
    o1 = f(x)              # main
    o2 = f(mask(x,r)) # sub (mask ratio: r)
    loss1 = CE(o1, label) / 2
    loss2 = CE(o2, softmax(o1.detach())) / 2
    (loss1+loss2).backward()
    optimizer.step()
```

https://arxiv.org/pdf/2306.11339 (CVPR 2025)

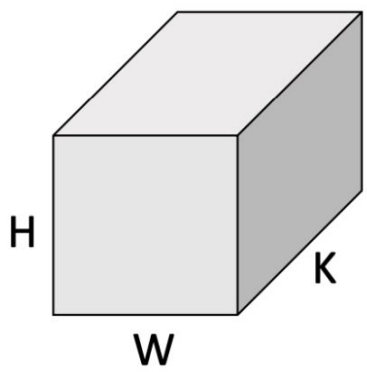
M | ROBOTICS

Class Activation Mapping (CAM)

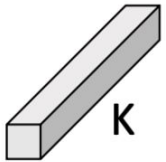


https://openaccess.thecvf.com/content_cvpr_2016/papers/Zhou_Learning_Deep_Features_CVPR_2016_paper.pdf

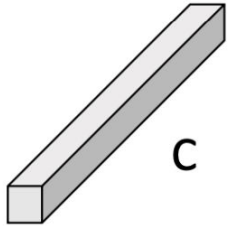
Class Activation Mapping (CAM)



Global Average Pooling



Fully Connected Layer, weights $w \in \mathbb{R}^{K \times C}$



Last layer CNN features:
$f \in \mathbb{R}^{H \times W \times K}$

Pooled features:
$F \in \mathbb{R}^{K}$

Class Scores:
$S \in \mathbb{R}^{C}$

$$F_k = \frac{1}{HW} \sum_{h,w} f_{h,w,k} \qquad S_c = \sum_k w_{k,c} F_k = \frac{1}{HW} \sum_k w_{k,c} \sum_{h,w} f_{h,w,k}$$

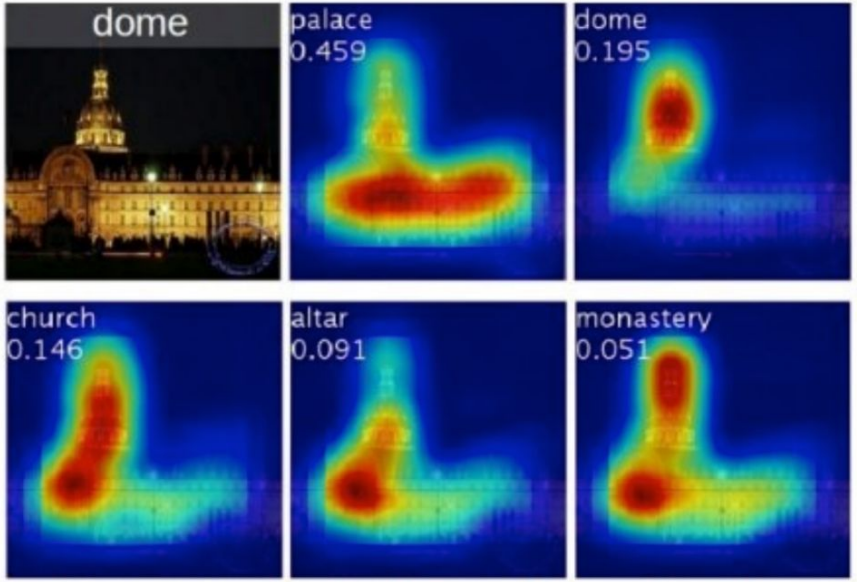
$$= \frac{1}{HW} \sum_{h,w} \sum_k w_{k,c} f_{h,w,k}$$

Class Activation Maps:
$M \in \mathbb{R}^{C,H,W}$

$$M_{c,h,w} = \sum_k w_{k,c} f_{h,w,k}$$

https://openaccess.thecvf.com/content_cvpr_2016/papers/Zhou_Learning_Deep_Features_CVPR_2016_paper.pdf

Class Activation Mapping (CAM)



Class activation maps of top 5 predictions

Class activation maps for one object class

https://openaccess.thecvf.com/content_cvpr_2016/papers/Zhou_Learning_Deep_Features_CVPR_2016_paper.pdf

Problem: Can only apply to last conv layer

M | ROBOTICS

Grad-CAM:
Gradient-Weighted Class Activation Mapping

1. Pick any layer, with activations $A \in \mathbb{R}^{H \times W \times K}$
2. Compute gradient of class score S_c with respect to A:

$$\frac{\partial S_c}{\partial A} \in \mathbb{R}^{H \times W \times K}$$

3. Global Average Pool the gradients to get weights $\alpha \in \mathbb{R}^{K}$:

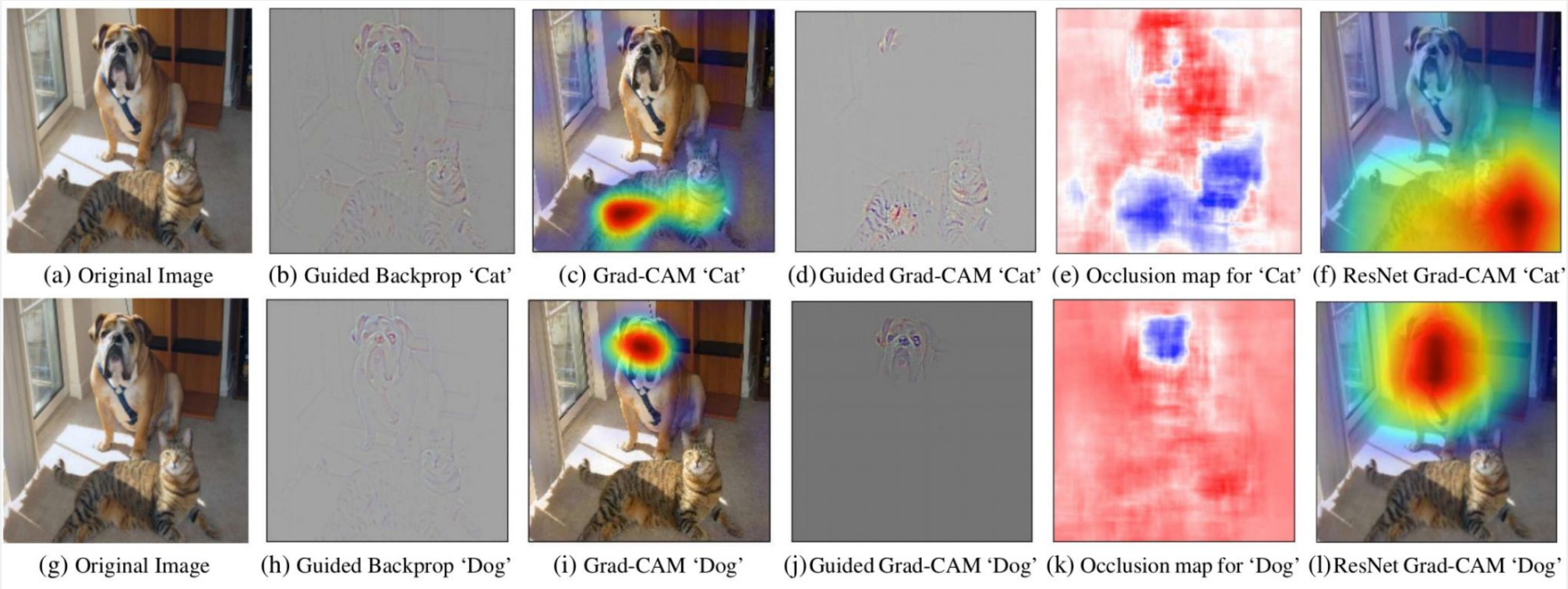
$$\alpha_k = \frac{1}{HW} \sum_{h,w} \frac{\partial S_c}{\partial A_{h,w,k}}$$

4. Compute activation map $M^c \in \mathbb{R}^{H,W}$:

$$M^c_{h,w} = ReLU\left(\sum_k \alpha_k A_{h,w,k}\right)$$

https://arxiv.org/pdf/1610.02391

Grad-CAM



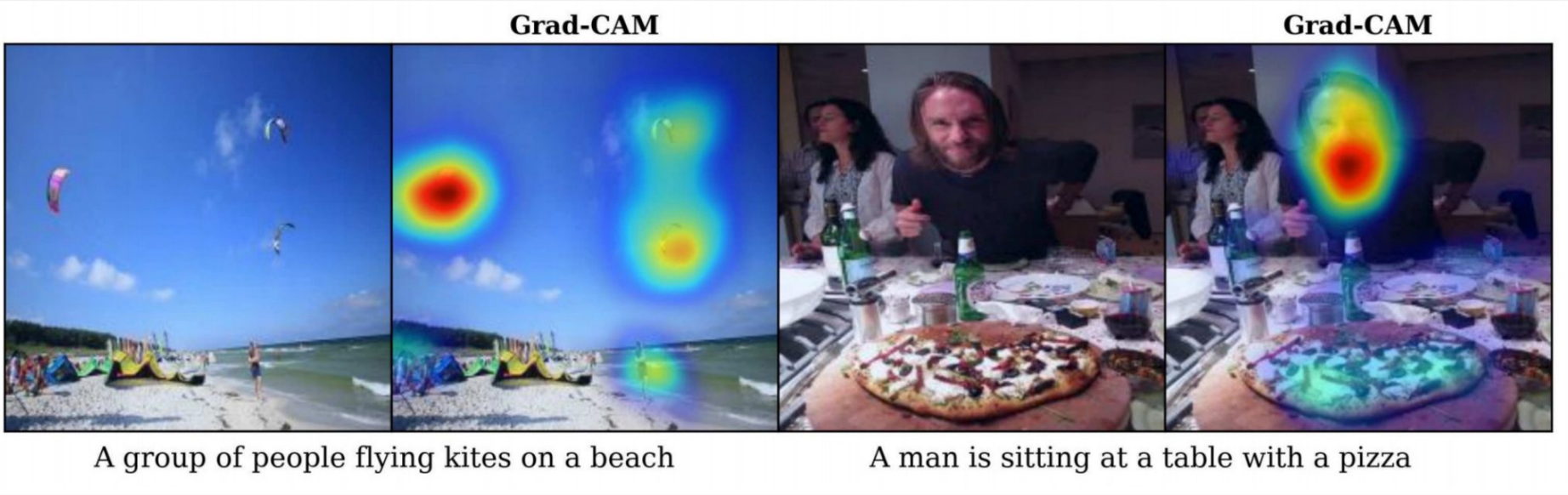
https://arxiv.org/pdf/1610.02391

M | ROBOTICS

Grad-CAM

https://github.com/jacobgil/pytorch-grad-cam

Grad-CAM for Image Captioning



https://arxiv.org/pdf/1610.02391

More methods:

- Guided backprop,
- Gradient Ascent
- ...

M | ROBOTICS

Vision-Language Models

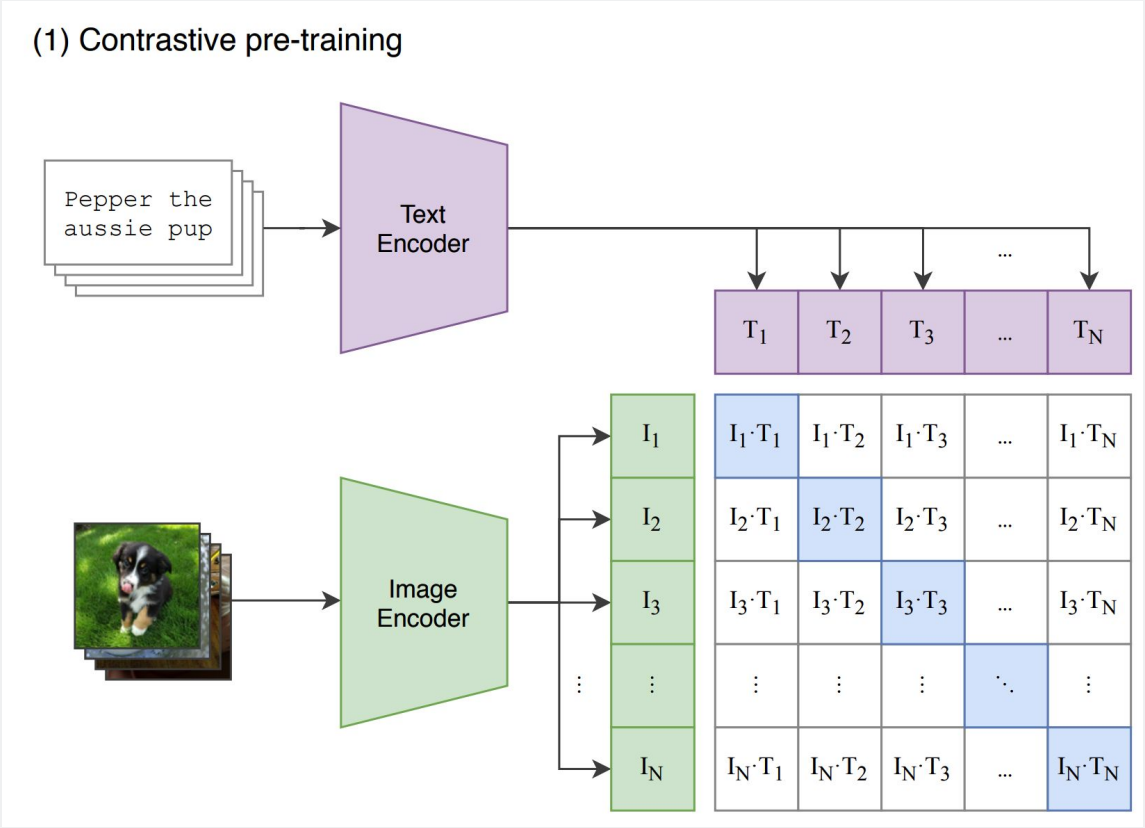
CLIP (2021)

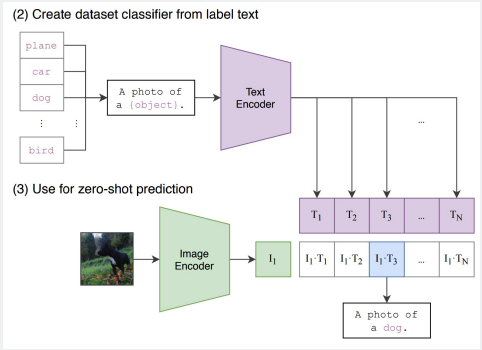
Cited by 32077

https://openai.com/index/clip/

https://arxiv.org/pdf/2103.00020

Contrastive Language-Image Pre-training





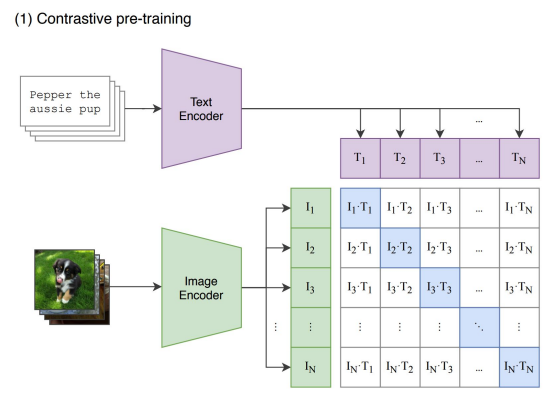
M | ROBOTICS

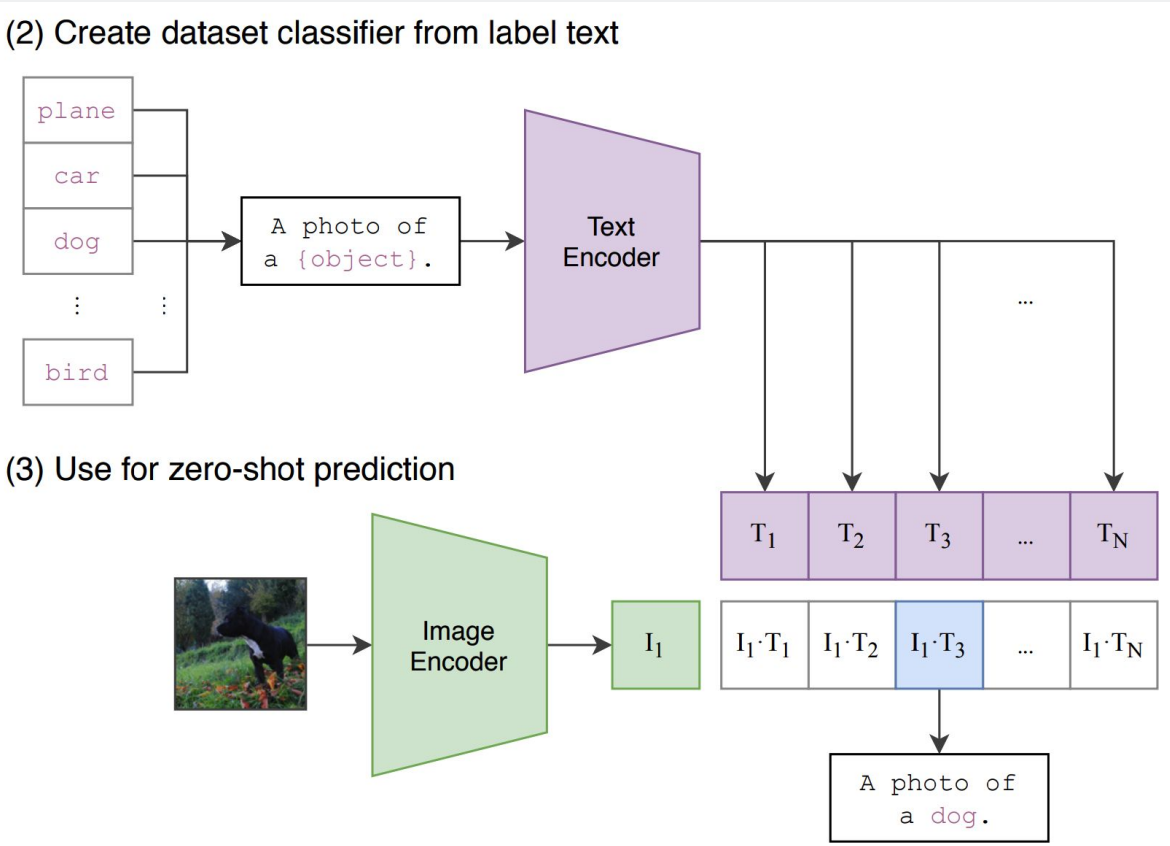
CLIP

https://openai.com/index/clip/

https://arxiv.org/pdf/2103.00020

Contrastive Language-Image Pre-training





CLIP

Food101

guacamole (90.1%) Ranked 1 out of 101 labels

✓ a photo of **guacamole**, a type of food.

× a photo of **ceviche**, a type of food.

× a photo of **edamame**, a type of food.

× a photo of **tuna tartare**, a type of food.

× a photo of **hummus**, a type of food.

Youtube-BB

airplane, person (89.0%) Ranked 1 out of 23 labels

✓ a photo of a **airplane**.

× a photo of a **bird**.

× a photo of a **bear**.

× a photo of a **giraffe**.

× a photo of a **car**.

```
# image_encoder - ResNet or Vision Transformer
# text_encoder  - CBOW or Text Transformer
# I[n, h, w, c] - minibatch of aligned images
# T[n, l]       - minibatch of aligned texts
# W_i[d_i, d_e] - learned proj of image to embed
# W_t[d_t, d_e] - learned proj of text to embed
# t             - learned temperature parameter

# extract feature representations of each modality
I_f = image_encoder(I) #[n, d_i]
T_f = text_encoder(T)  #[n, d_t]

# joint multimodal embedding [n, d_e]
I_e = l2_normalize(np.dot(I_f, W_i), axis=1)
T_e = l2_normalize(np.dot(T_f, W_t), axis=1)

# scaled pairwise cosine similarities [n, n]
logits = np.dot(I_e, T_e.T) * np.exp(t)

# symmetric loss function
labels = np.arange(n)
loss_i = cross_entropy_loss(logits, labels, axis=0)
loss_t = cross_entropy_loss(logits, labels, axis=1)
loss   = (loss_i + loss_t)/2
```

Figure 3. Numpy-like pseudocode for the core of an implementation of CLIP.

CLIP

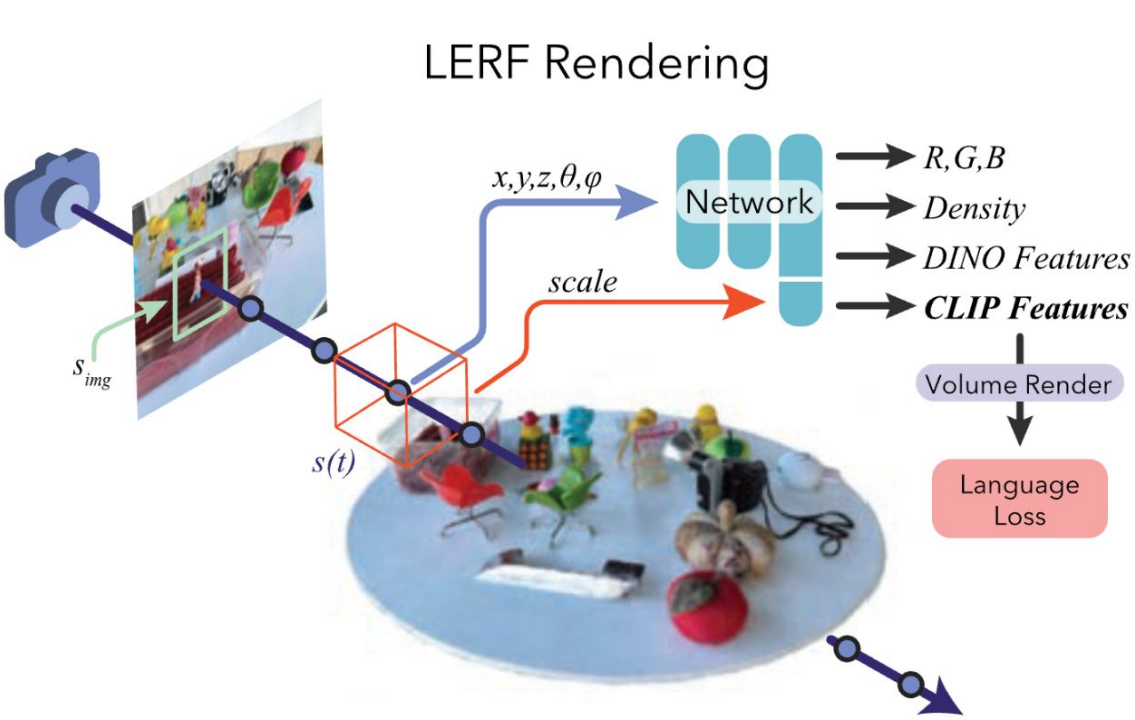
(+) Efficient. Can be trained on noisy data
(+) more flexible and general than other ImageNet models (at the time)
(-) struggles on more **abstract** or complex tasks, such as counting the number of objects, predicting how close the nearest car is in the photo
(-) struggles on very **fine-grained** classification (e.g., between models of car or flower)

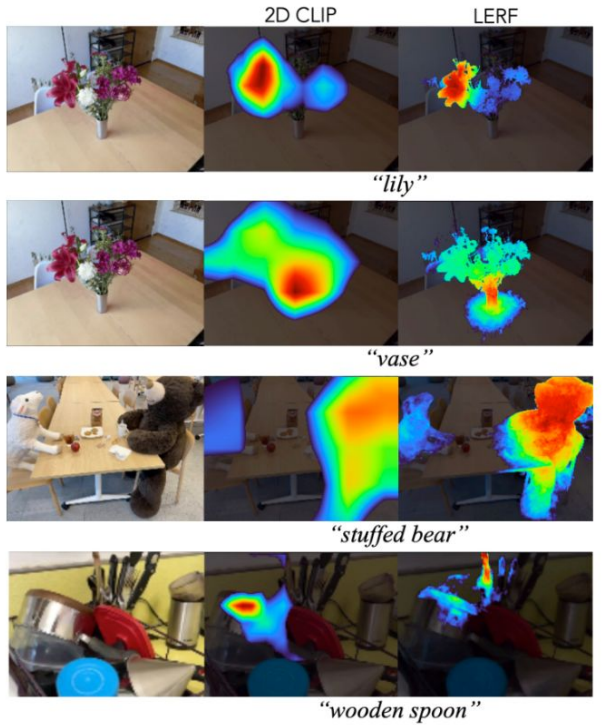
Prompt template 

LERF (2023)

https://www.lerf.io/

Language Embedded Radiance Fields

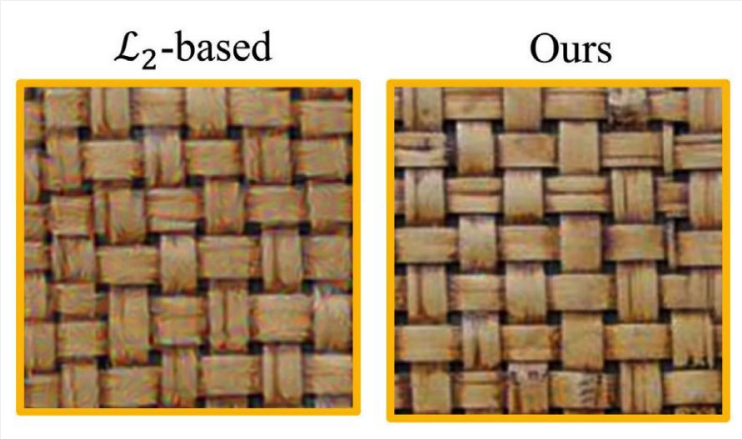


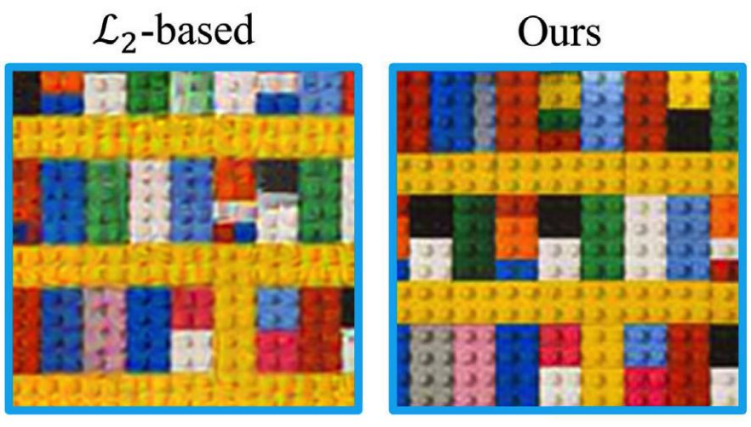


M | ROBOTICS

Neural Texture Synthesis

(CVPR 2023)





https://openaccess.thecvf.com/content/CVPR2023/papers/Zhou_Neural_Texture_Synthesis_With_Guided_Correspondence_CVPR_2023_paper.pdf

Diffusion Model for Style Transfer

(CVPR 2024)

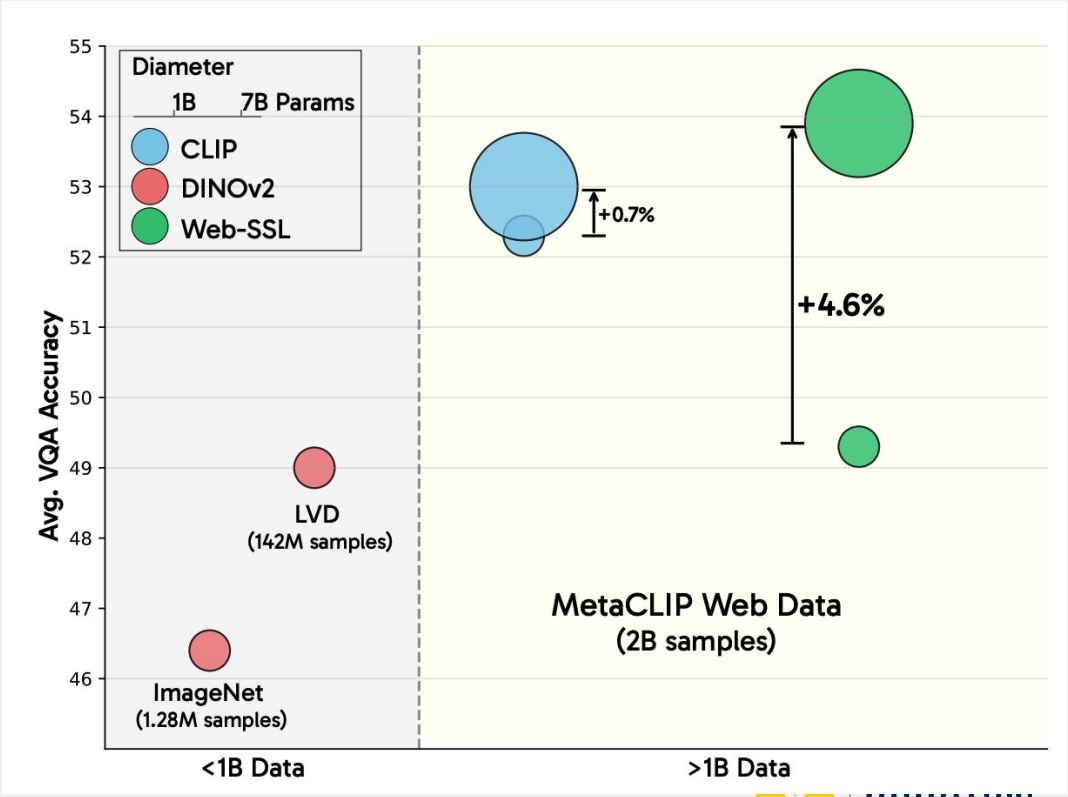
https://openaccess.thecvf.com/content/CVPR2024/papers/Chung_Style_Injection_in_Diffusion_A_Training-free_Approach_for_Adapting_Large-scale_CVPR_2024_paper.pdf

Scaling Language-Free Visual Representation Learning

https://arxiv.org/pdf/2504.01017
https://davidfan.io/webssl/

Released April 1, 2025

"Pure self-supervised learning such as DINO can beat CLIP-style supervised methods on image recognition tasks because SSL performance **scales well** with architecture size and dataset size."

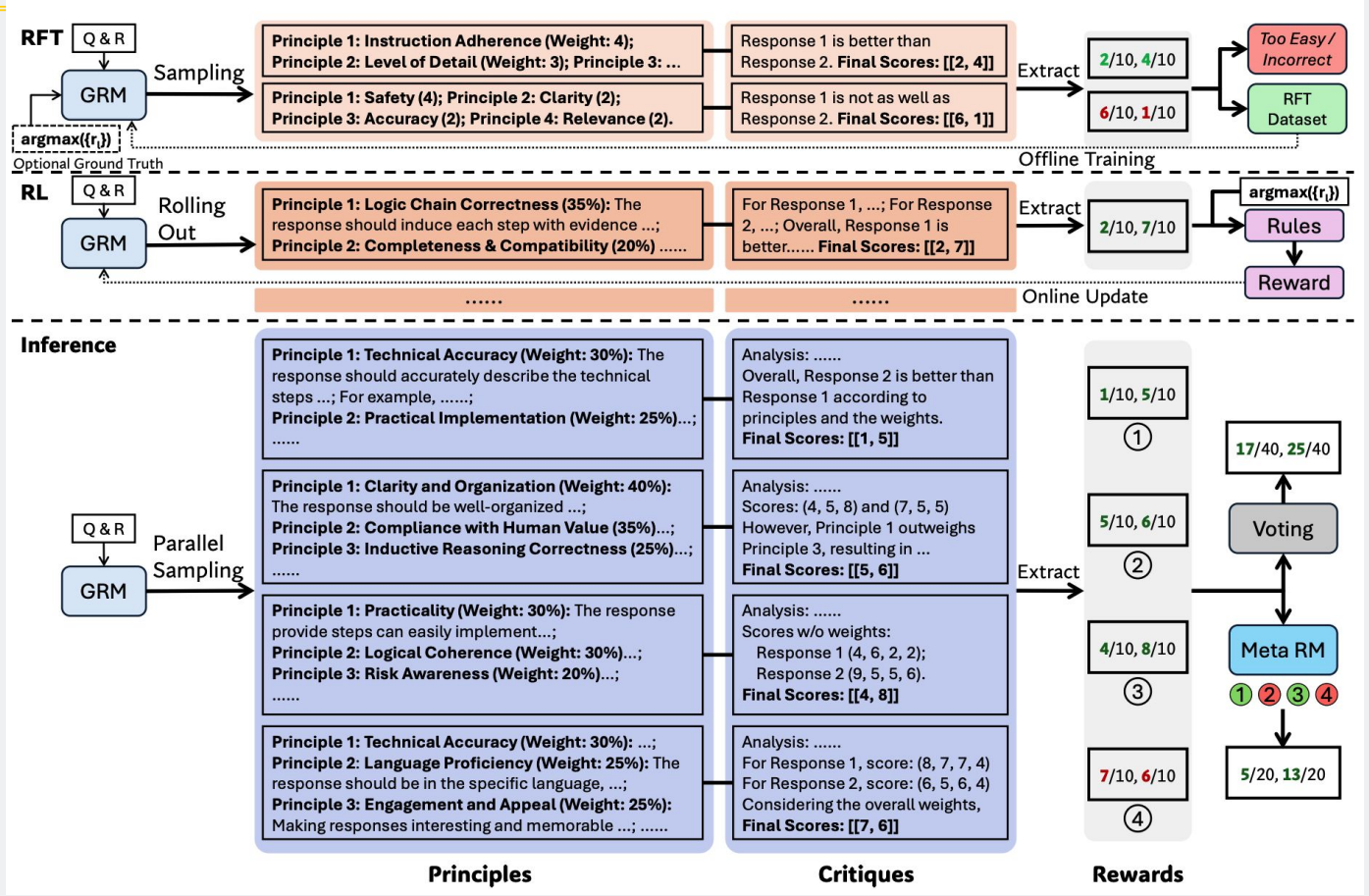


DeepSeek-GRM

https://arxiv.org/pdf/2504.02495

Released
April 3, 2025

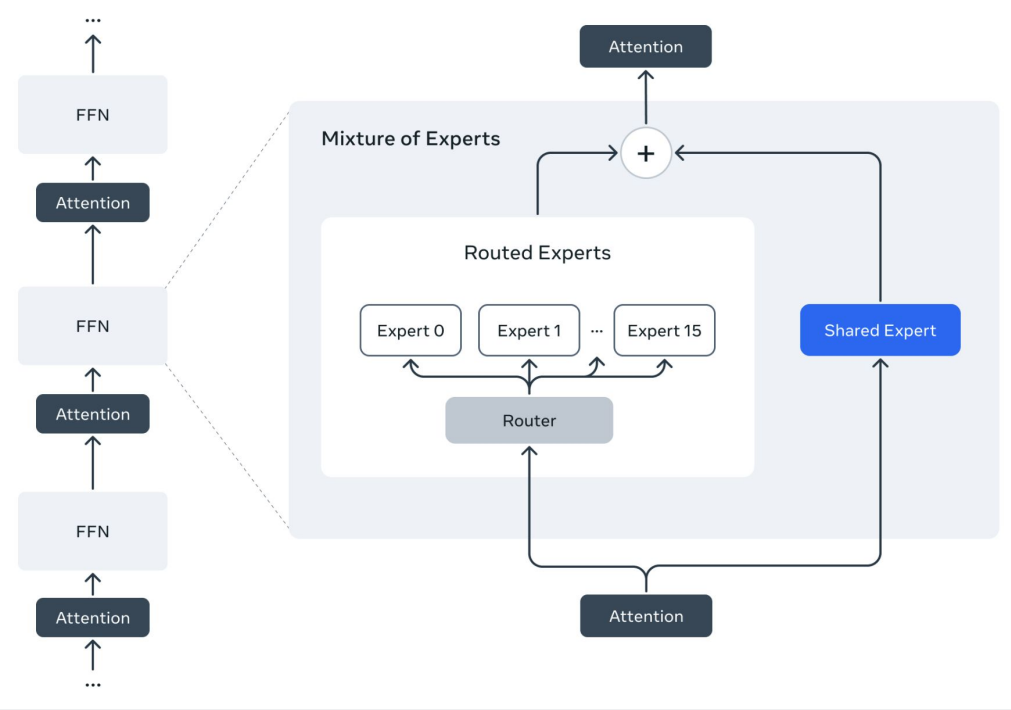
- Generative reward modeling
- Self-Principled Critique Tuning



Llama 4

https://ai.meta.com/blog/llama-4-multimodal-intelligence/

Released April 5, 2025



Llama 4 Behemoth

288B active parameter, **16** experts
2T total parameters

The most intelligent teacher model for distillation

Llama 4 Maverick

17B active parameters, **128** experts
400B total parameters

Native multimodal with **1M** context length

Llama 4 Scout

17B active parameters, **16** experts
109B total parameters

Industry leading **10M** context length
Optimized inference

Available

ROBOTICS

“Do two AI Scientists Agree?”

https://arxiv.org/pdf/2504.02822

Released April 3, 2025

“allow a single neural network to learn diverse theories across multiple physical systems”

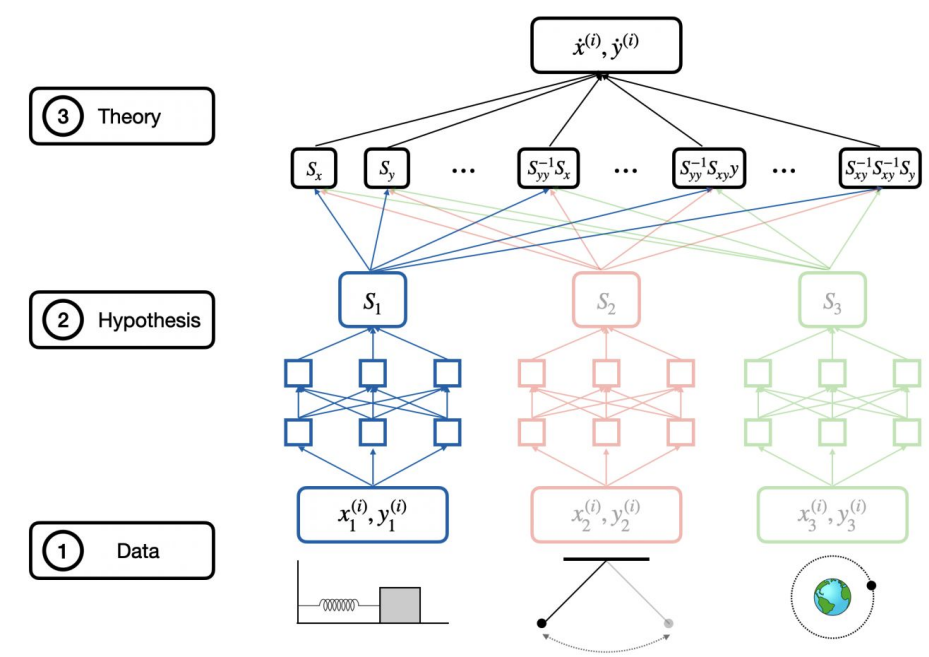


FIG. 2. The MASS (Multi-physics AI Scalar Scientist) network.

“strong correlation”
“scientists can learn different theories”
“Modify new theories to fit new data”

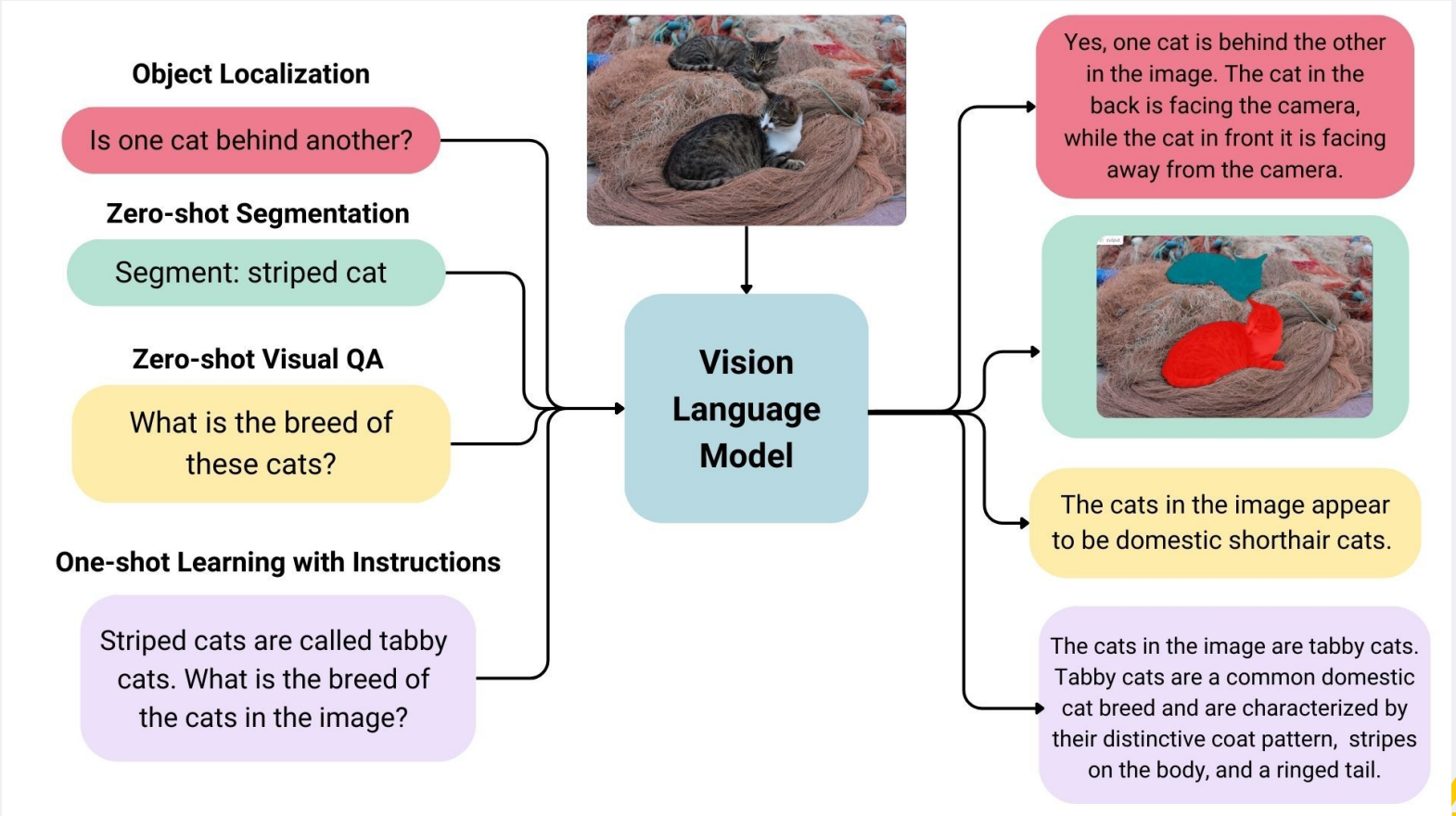
M | ROBOTICS

Vision-Language Models (summary)

https://www.nvidia.com/en-us/glossary/vision-language-models/ (NVIDIA)
https://huggingface.co/blog/vlms (Hugging Face)
https://www.ibm.com/think/topics/vision-language-models (IBM)

An Introduction to VLM (2024) https://arxiv.org/pdf/2405.17247
Survey of SOTA VLMs (2025) https://arxiv.org/pdf/2501.02189

Vision-Language Models (summary)



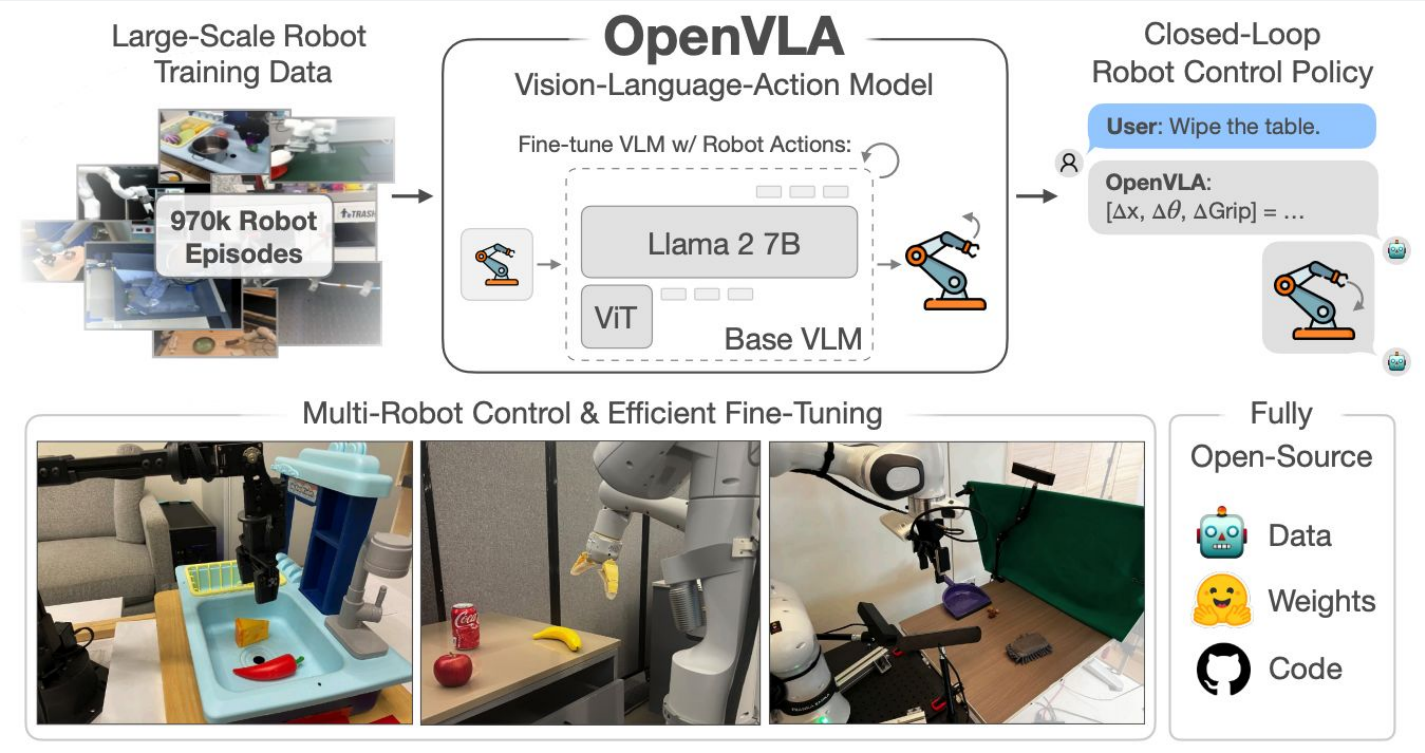
https://huggingface.co/blog/vlms

Open VLA

Open-Source Vision-Language-Action Model

https://arxiv.org/pdf/2406.09246 (Sept. 2024)

https://openvla.github.io/



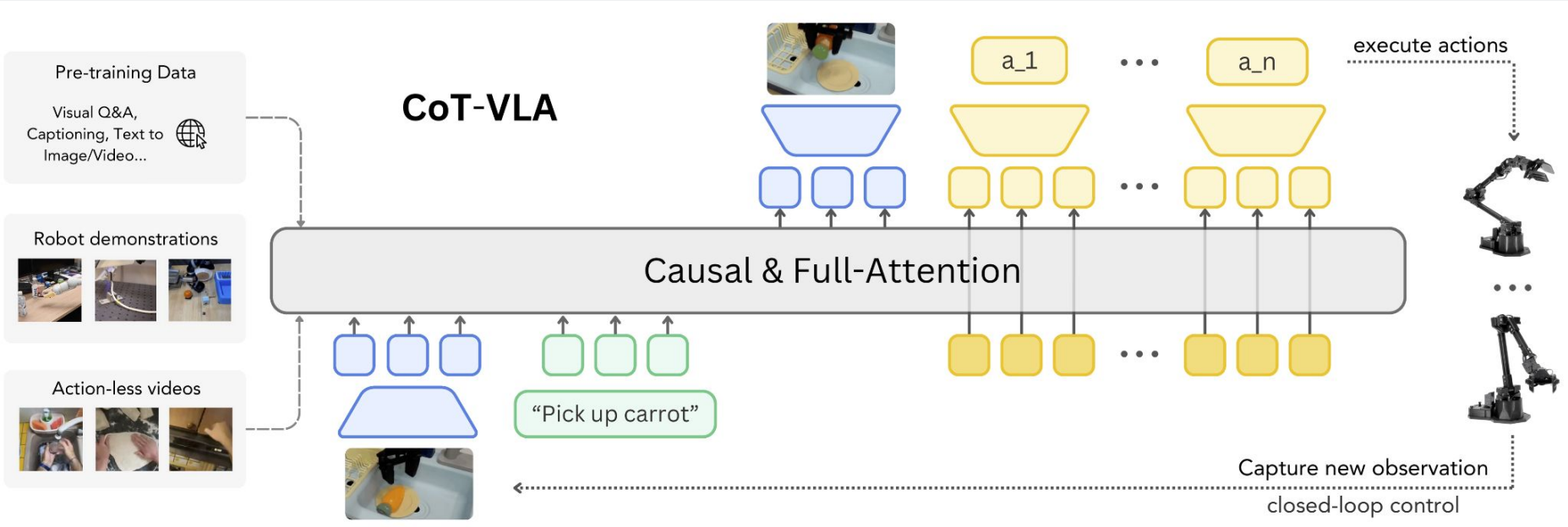
obtain robust, generalizable policies for visuomotor control

M | ROBOTICS

COT-VLA

Visual Chain-of-Thought Reasoning for
Vision-Language-Action Models

https://arxiv.org/pdf/2503.22020 (March 27, 2025) https://cot-vla.github.io/



M | ROBOTICS

COT-VLA

Visual Chain-of-Thought Reasoning for
Vision-Language-Action Models

https://arxiv.org/pdf/2503.22020 (March 27, 2025)

https://cot-vla.github.io/

Figure 3. **Hybrid attention mechanism in CoT-VLA**. We use causal attention for image or text generation and full attention for action generation. $[x]$, $[\theta]$ and $[g]$ are special tokens for parallel decoding of actions.

Trained on:

negative log-likelihood loss for visual tokens

+

https://arxiv.org/pdf/2409.04429#page=5.5

cross-entropy loss for action predictions

M | ROBOTICS

Logistics

Remaining Lectures (Plan)

Wednesday April 9: multi-sensor fusion

Monday April 14: Prof. Kaveh Fathian (https://cs.mines.edu/project/fathian-kaveh/)

Wednesday April 16: Prof. Karthik Desingh (https://cse.umn.edu/cs/karthik-desingh)

Monday April 21: Prof. Glen Chou (https://glenchou.github.io/)

Final Project report requirements

Total: 18%

IEEE Format (overleaf recommended)
https://www.ieee.org/conferences/publishing/templates.html

1. Report: IEEE paper format (title, your names, abstract, introduction, related work, method, experiments and results, discussion and conclusion, references)

Paper Reproduction (can be a summary of what the original paper did and what you did to reproduce)	+	Algorithmic Extension (emphasize **your novel** contributions)

2. Code: Link to code repo or zip file
3. Video/website: Please also submit **either a video or website** link introducing your work.
See previous website examples: https://deeprob.org/w24/reports/ If you would like deeprob to host your website, please reach out to Anthony.

M | ROBOTICS

Reminders

- **Lightning Talk feedback released via Canvas**
- **Final Project showcase (*bonus for live robot demo!*)**
  April 22, 2025
- **Final Report DUE**
  April 28, 2025
- **Canvas Quiz** will be released

- **Invited Lectures:** There will be attendance checks and canvas quizzes associated with the invited lectures - please attend! We will also send out message to gather questions/feedback for our speakers

- **Course Eval:** please type the text "Done" on canvas so that we know you filled in the course eval. It will count as **0.5 points** in your participation score. Thank you very much in advance for your feedbacks!

M | ROBOTICS