Adaptive Testing of Computer Vision Models

MI2 Seminar

Mikołaj Spytek

November 13th, 2023

Authors

Irena Gao
(Stanford University)

Gabriel Ilharco
(University of Washington)

Scott Lundberg
(Microsoft Research)

Marco Tulio Ribeiro
(Microsoft Research, now Google Deepmind)

International Conference on Computer Vision 2023.

What is the use case?

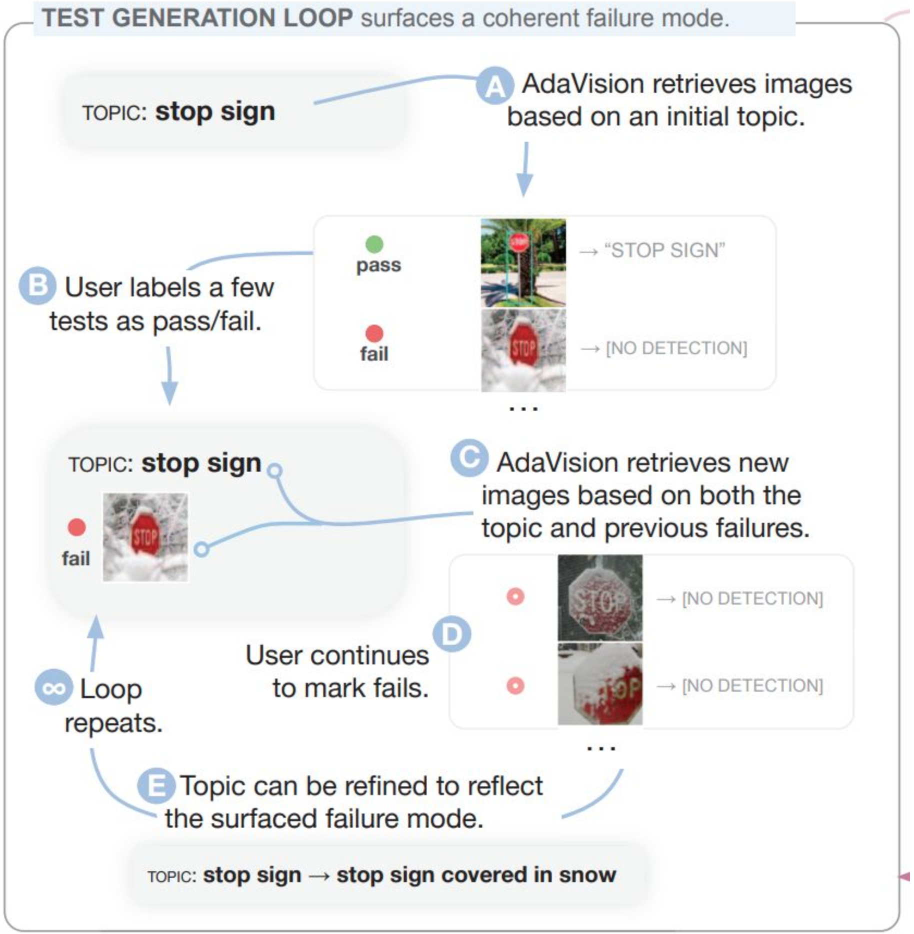
- **Finding modes of unexpected failure of computer vision models:**
  - **Increasing performance** - models can be fine-tuned using a targeted collection of difficult examples
  - **Deployment** - decision-makers can decide if their models are safe and fair to deploy (e.g., holding off deployment of autonomous driving, when computer vision model doesn't perform well in unusual weather)

Previous approaches

- **Clustering errors** - some approaches cluster errors from the evaluation set in different approaches. These methods sadly lead to incoherent groups - there is no easy response. They are also limited by samples from the test set.
- **Human-in-the-loop** - these approaches are popular in NLP (which is understandable, as people can generate examples for NLP models), however, there are no established frameworks for the Computer Vision modality.

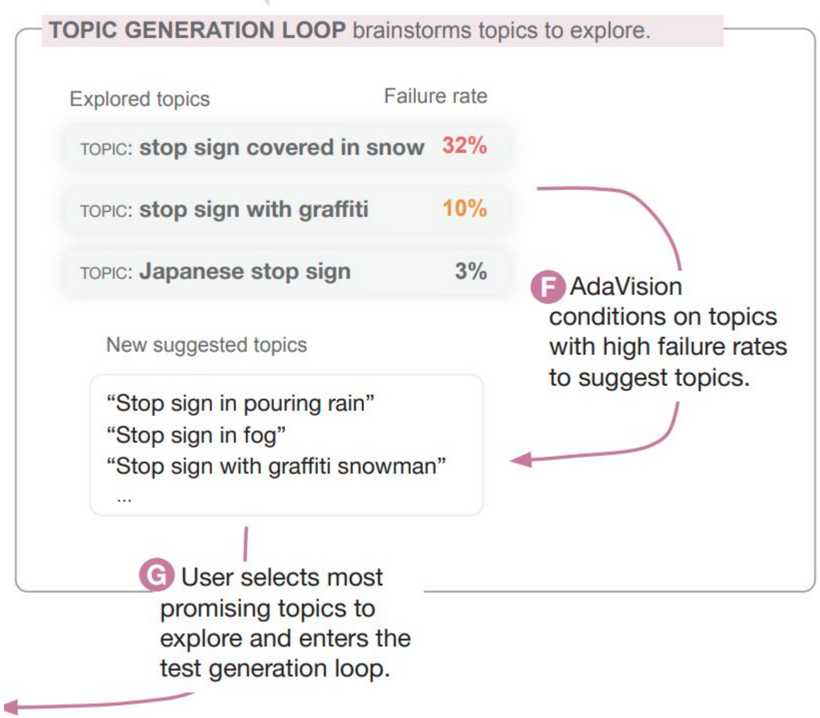
Main idea

First, a **general topic** is selected. Then images relevant to this topic are selected from the **LAION-5B dataset** are retrieved with the use of **CLIP embeddings**. Model **predictions are obtained** for these images and **users label** if these predictions are correct. Then, the general **topic can be refined.**



Obligatory LLM interlude

The solution proposed by authors also includes a module, which **makes it easy** to generate more **specific topics**, with which models can have problems with the use of Large Language Models.

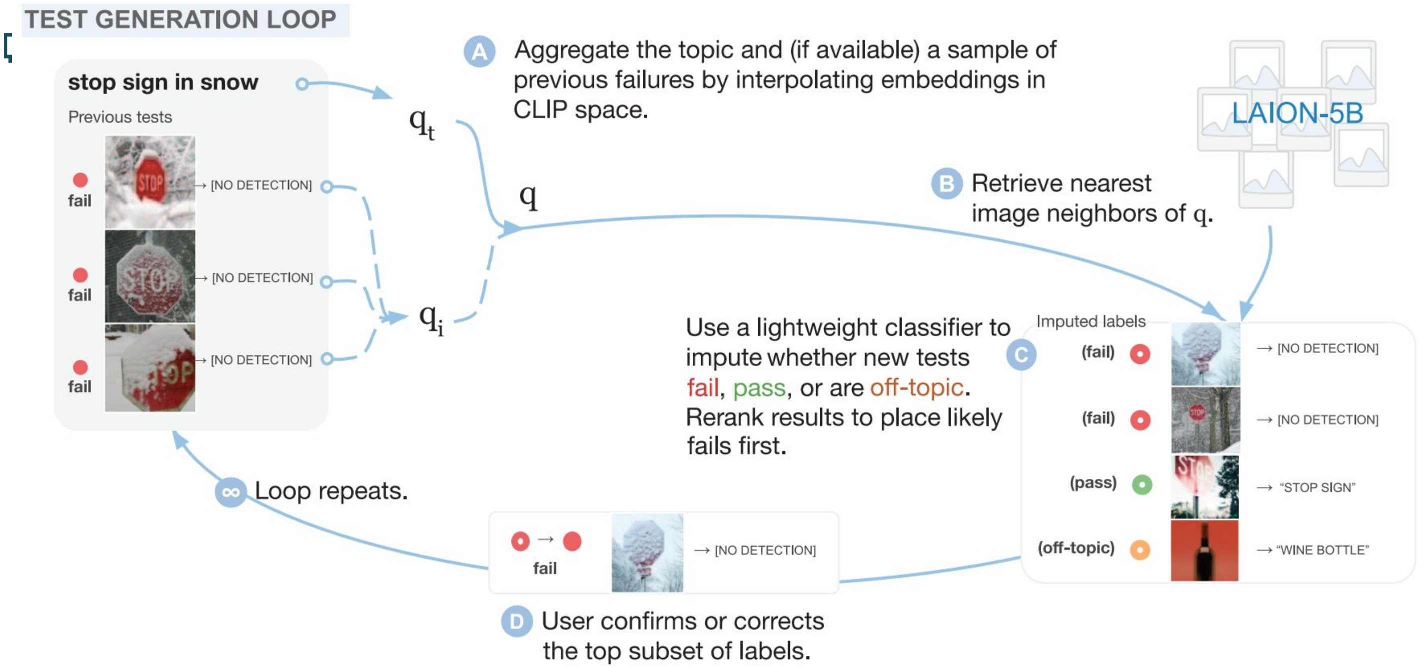


Definitions

- **model - m**; classification, object detection and captioning models were considered,
- **observation - x**; only images were considered in this paper,
- **test** - the observation **x** and expected behavior of model **m** on **x**
- **test failure -** an observation **x**, for which **m(x)** doesn't match expectations
- **topic** - a set of **tests**, whose images are united by a human-understandable concept
- **bug** - a topic with **high failure rate**
- **P(X|T)** - the distribution of images given topics

$$\mathbb{E}_{x\sim\mathbf{P}(\mathbf{X}|\mathbf{t})}\,[\mathrm{test}(x)\ \mathrm{fails}] \gg \mathbb{E}_{x\sim\mathbf{P}(\mathbf{X})}\,[\mathrm{test}(x)\ \mathrm{fails}]$$

Test generation loop



Test generation algorithm

Algorithm 1: Iteration of the test generation loop.

Input: Textual topic description z, previously labeled tests $\mathcal{D} = \{(x, m(x), y)\}$, previous off-topic tests $\mathcal{D}_{\text{off-topic}}$

Compute $q_t \leftarrow \text{CLIP}(z)$ ▷ Figure 2A
if $|\mathcal{D}| > 0$ **then**
  Sample $x_1, x_2, x_3 \sim \text{Categorical}(|\mathcal{D}|, p_j)$, where p_j is computed according to the text in A.1
  Aggregate $q_i \leftarrow \sum_k \beta_k \cdot \text{CLIP}(x_k)$, with $\beta \sim \text{Dirichlet}(1, 1, 1)$
  Set $q \leftarrow \text{slerp}(q_t, q_i, r)$, with $r \sim \text{Uni}(0, 1)$
else
  Set $q \leftarrow q_t$
end

Retrieve approximate nearest neighbors of q from LAION-5B ▷ Figure 2B
Exclude retrievals whose CLIP image embeddings have cosine similarity > 0.9 with any previous test $x \in \mathcal{D}$
Collect model outputs for all retrieved images to obtain new collection of tests $\mathcal{S} \leftarrow [(\tilde{x}, m(\tilde{x}))]$

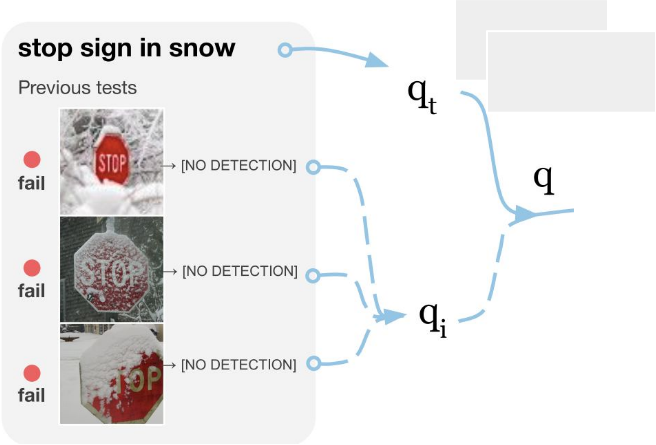
if $|\mathcal{D}| > 0$ **then**
  ▷ Figure 2C
  Train a lightweight classifier f on previously labeled tests $\mathcal{D}$ as described in A.1
  Sort $\mathcal{S}$ according to $f(\tilde{x})$ for $\tilde{x} \in \mathcal{S}$, placing predicted fails far from the decision boundary first, and predicted passes far from the decision boundary last Update $\mathcal{S}$ to contain $(\tilde{x}, m(\tilde{x}), f(x))$, so that we can display the imputed label to the user
  Train a second lightweight classifier $f_{\text{off-topic}}$ to differentiate between previous in-topic tests $\mathcal{D}$ and previous off-topic tests $\mathcal{D}_{\text{off-topic}}$
  Place tests $\tilde{x} \in \mathcal{S}$ for which $f_{\text{off-topic}}(x)$ predicts "off-topic" at the end of $\mathcal{S}$
end
return sorted $\mathcal{S}$ to the user for confirmation / correction. ▷ Figure 2D

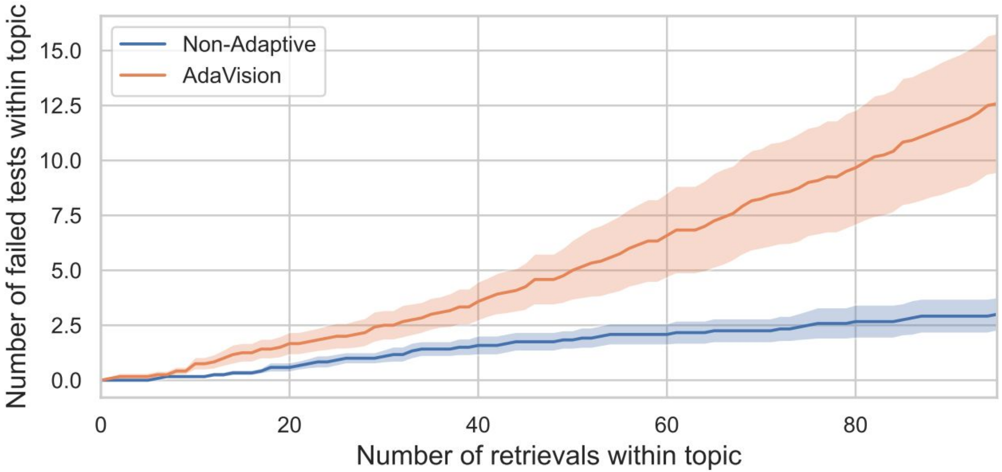
Topic correction - *let's use some LLMs*

- List some unexpected places to see a { LABEL }
- List some places to find a { LABEL }
- List some other things that you usually find with a { LABEL }
- List some artistic representations of a { LABEL }
- List some things that can be made to look like a { LABEL }
- List some types of { LABEL } you wouldn't normally see
- List some dramatic conditions to photograph a { LABEL }
- List some conditions a { LABEL } could be in that would make it hard to see
- List some things that are the same shape as a { LABEL }
- List some { LABEL } that are a different color than you would expect

Does adaptivity help?

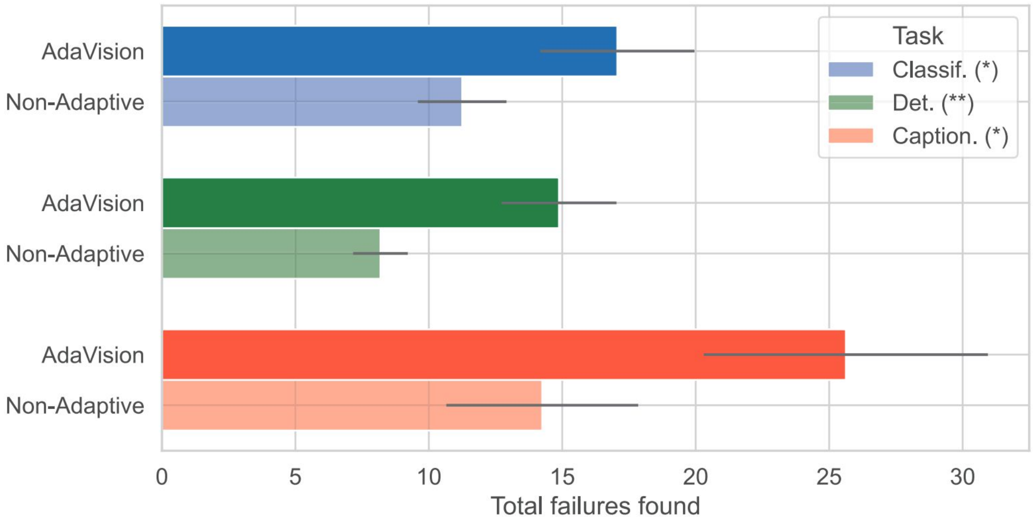
Adapting, for the case of this experiment means **averaging** the **embeddings** of previous failure cases, and **combining** them with the original **textual topic description**.





User study

- **Three** different tasks (classification, object detection and image captioning)
- Classification - **ViT-H/14** on **banana** and **broom** categories.
- Object detection - **Google Cloud Vision API** on **bicycle** and **stop sign** categories.
- Captioning - **OFA-Huge** on **kitchen** and **elementary school** scenes. (Failures are captions which would mislead a visually impaired users)



- **40 participants** from academia and industry, after an **ML course**
- **20 minutes** for each round
- priority on finding **as many bugs as possible**
- **84.6%** of users say they couldn't have found these bugs using existing analysis tools

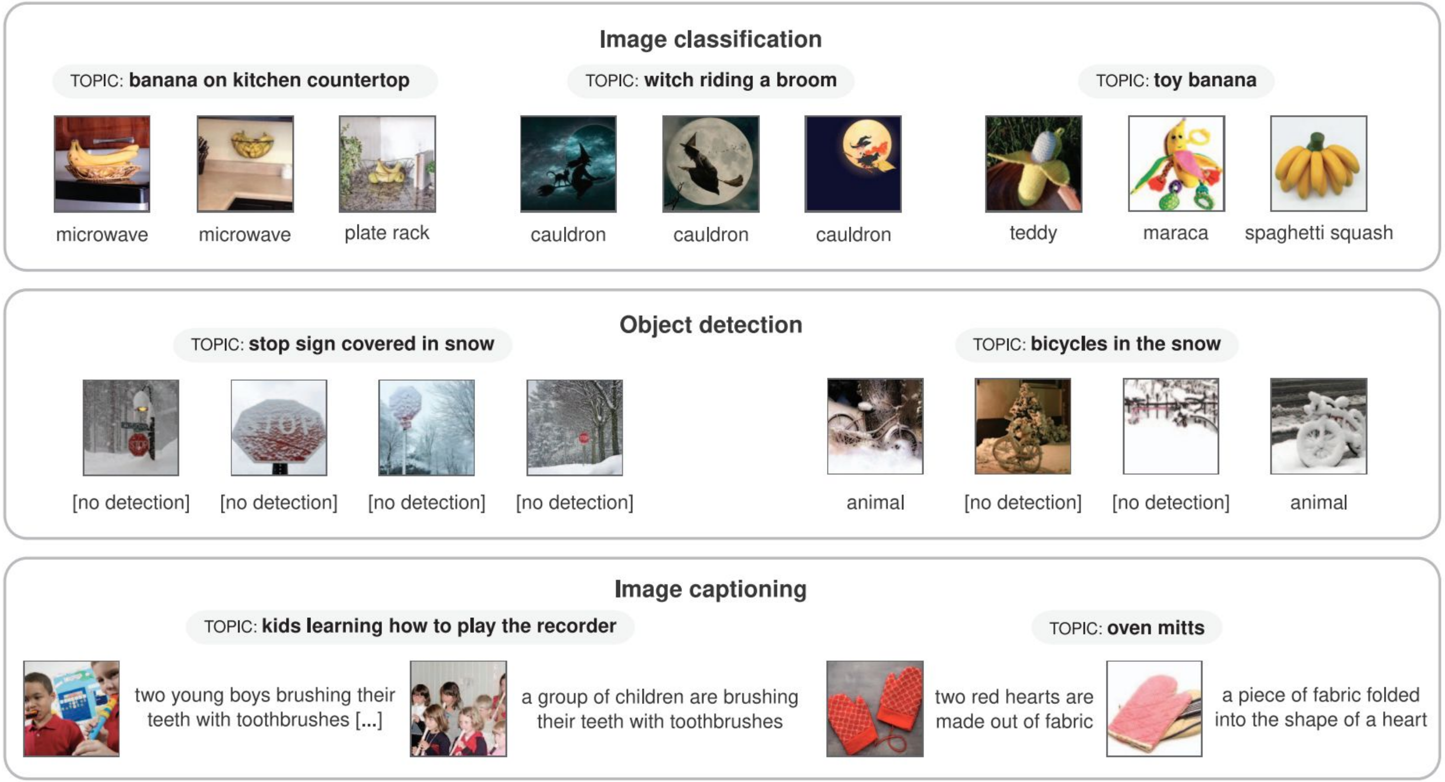


Image classification
TOPIC: banana on kitchen countertop
microwave
microwave
plate rack
TOPIC: witch riding a broom
cauldron
cauldron
cauldron
TOPIC: toy banana
teddy
maraca
spaghetti squash
Object detection
TOPIC: stop sign covered in snow
[no detection]
[no detection]
[no detection]
[no detection]
TOPIC: bicycles in the snow
animal
[no detection]
[no detection]
animal
Image captioning
TOPIC: kids learning how to play the recorder
two young boys brushing their teeth with toothbrushes [...]
a group of children are brushing their teeth with toothbrushes
TOPIC: oven mitts
two red hearts are made out of fabric
a piece of fabric folded into the shape of a heart

Comparison with an automatic method

- **DOMINO** is a method that **clusters** validation set **errors** and describes them with automatically **generated captions**.
- Tests for **6 categories**: banana, broom, candle, lemon, sandal, wine bottle
- The **average failure rate** indicates the percentage of test failures in the images retrieved from **LAION-5B,** when querying for the **captions** generated by various methods.

Model	Method	Avg failure rate
ViT-H/14	`a photo of {y}`	1.33
	ImageNet	11.47
	DOMINO (BERT)	8.6
	DOMINO (OFA)	7.33
	ADAVISION	**28.47**
ResNet50	`a photo of {y}`	15.7
	ImageNet	23.67
	DOMINO (BERT)	20.44
	DOMINO (OFA)	25.45
	ADAVISION	**56.93**

Fine-tuning on failures

	AdaVision Topics			ImageNet	Avg across OOD Eval Sets	
Model	**Treatment Topics**		**Control Topics**	**Overall**	**Treatment Classes**	**Overall**
	LAION-5B	*Google*				
Before finetuning	72.6	76.7	91.3	88.4	78.0	77.7
Finetuning with `an image of {y}`	82.5 (0.9)	82.9 (0.6)	90.8 (0.3)	**88.5 (0.0)**	82.1 (0.6)	78.0 (0.1)
Finetuning with AdaVision tests	**91.2 (0.5)**	**90.6 (0.6)**	**91.9 (0.2)**	88.4 (0.0)	**84.0 (0.2)**	**78.2 (0.0)**

- Fine-tuning the **ViT-H/14** model on **600 images** (6 categories x 5 topics x 20 tests)
- Improved performance on **bugs**
- Maintained **in-distribution** performance
- Better performance on out-of-distribution

Limitations

- The **LAION-5B** has good **coverage** for everyday scenes, but is not appropriate for specific domains (medical, satellite images, etc.).
- The performance of **CLIP also deteriorates** in special domains that are not covered in generic databases.
- The experiments focus on a specific set of **labels with high accuracy** to begin with and only a **handful of models**.
- **Classification is easy** to evaluate as we can say if the test **passes or fails.** For other task it is **not clear what constitutes a failure**, as it most often depends on the specific use case.

Conclusions

- AdaVision is a **human-in-the-loop process** for testing computer vision models.
- **Human feedback helps** identify and improve **coherent topics**, where models fail (bugs).
- Experiments show that **AdaVision improves the discovery of bugs** compared to other methods.
- Fine-tuning on the discovered bugs **boosts performance in the problematic failure modes**, while keeping in-distribution performance.

Thank you!

Questions?

November 13th, 2023