Tips to reduce the pain of image labeling

Speaker: Sonia Tabti, PhD

DATAQUITAINE

at Kedge Business School, 20/03/2025

Licence

CC BY-NC-ND

Who am I ? → Sonia Tabti

- 2024 - Present: Freelance AI specialist

AI Research
AI product Roadmap
Computer Vision
Problem solving

My clients

- 12+ years of expertise: Half corporate and half academic

PhD student → Researcher → Senior data scientist → AI team leader → Head of AI Research

That famous number you might all know about

→ 80 % of AI projects fail

→ It’s twice higher than other IT projects

Source: The Root Causes of Failure for Artificial Intelligence Projects and How They Can Succeed, J. Ryseff et al. (RAND), 2024

That famous number you might all know about

→ 80 % of AI projects fail

→ It's twice higher than other IT projects

Why is that ? Some of the main bottlenecks
• Data collection • Data labeling ← Today's focus ! • Deployment

Source: The Root Causes of Failure for Artificial Intelligence Projects and How They Can Succeed, J. Ryseff et al. (RAND), 2024

Outline

1. One of the main bottlenecks of AI projects
    a. Context
    b. Why is labeling so hard ?
2. Different ways to reduce the pain of image labeling
    a. Use a good interface
    b. Build a strong review methodology
    c. Some modeling strategies to ease the pain
3. Focus on Vision-Language Models to accelerate image labeling
    a. From open world models to VLMs
    b. Proposed semi-automated labeling workflow

Image labeling examples

What's labeling/annotating a data sample ?

It's the, often manual, process of assigning a label to data so that the model can be trained or evaluated on examples whose ground truth is known.

Task	Label type	Annotation process
Classification: Associate an image to a class	 - Dog ✅ - Cat	Select a class among predefined ones
Object detection: Locate objects in boxes and categorize them		Draw a bounding box (bbox) around objects of interest and select their category among a predefined list
Segmentation : Locate objects, compute their boundaries and categorize them		Draw polygons around the boundaries of objects of interest and select their category among a predefined list

Why is labeling so hard ?

- Not well understood, not anticipated
- Expansive
- Time consuming

- Limited data volume / the patterns we need to observe are rare

- Requires expertise
  - Good methodology
  - SMEs (Subject Matter Experts) must do the labeling in many cases
- Not easy to outsource
  - But if you do, write down very clear specifications

- Requires to use the right tools

Tip 1: Use a good labeling interface

For more:

- Efficient and collaborative work
- Ergonomy
- And many other features ...

Here are my two favorite open source tools to easily build labeling interfaces:

Label Studio

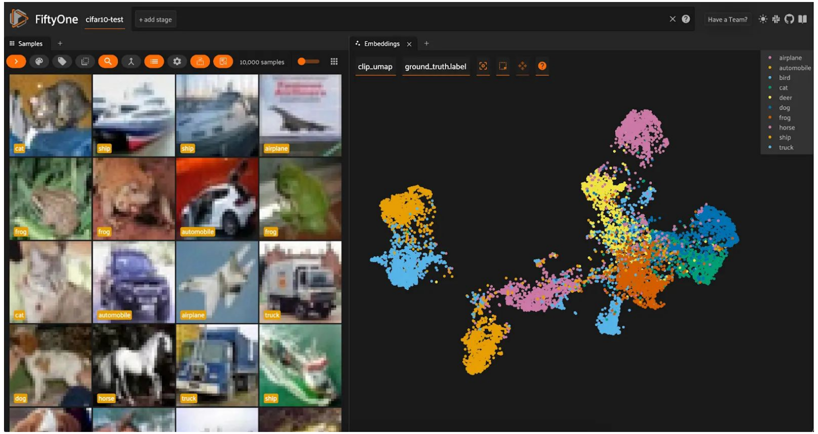
CVAT

Tip 2: Build a strong labeling review methodology

- Annotation → visual review
- Compute **stats** of the labels' distribution **regularly**
- Give **feedbacks** to the labeling team **regularly**
- Visualize the **embeddings** of the images (or bounding boxes) in 2D to spot obvious labeling mistakes
    - **How?:** You can do it yourself but I recommend to use this open-source tool:

FiftyOne

VOXEL51



Tip 3.1: modeling strategies that can help you

- Collect metadata to label the images
  - Eg: for defect detection, if reports listing the defects exist, use them
- And of course, use pretrained models - if relevant
  - Even better, check if an open-source model already exists and works for your use case
- Use data augmentation - if relevant

Tip 3.2: modeling strategies that can help you

Weakly supervised learning

→ Use less labeled data

- Unsupervised learning
  - Eg: anomaly detection
- Few shots learning
- Semi-supervised learning (SSL)

ANOMALiB

USB
Unified SSL Benchmark

Active learning

→ Improve model's performance iteratively as you label more data

→ Label in priority samples with higher uncertainty scores

Models combining text and images

They can be very helpful for:

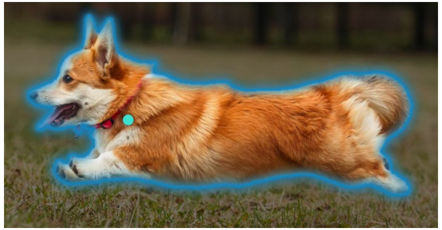
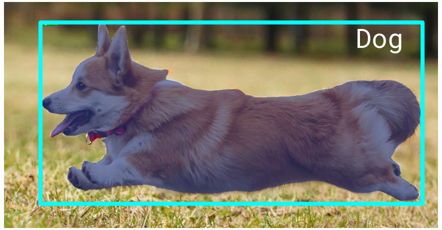
- Object detection
- Segmentation
- VQA (Visual Question Answering)
- OCR (Optical Character Recognition)
- Image captioning, …

→ You can fine tune them

→ But mostly you can use them to semi-automate data labeling and train an efficient model with this data to be deployed

We will talk more about them in the next slides …

From a foundation model to open world models 💚

Models' labeling power	SAM	Per-SAM	Grounded-SAM
Input	 Image + click or bounding-box	 Reference image + mask	 "Dog" Image + prompt
Output	 Mask of the selected object	 Mask for similar objects when applied on other images	 Bboxes and masks of the objects prompted

From a foundation model to open world models 💚

Models' labeling power

SAM → Per-SAM → Grounded-SAM

Input

SAM:

Image + click or bounding-box

Per-SAM:

Reference image + mask

Grounded-SAM:

"Dog"

Image + prompt

Output

SAM:

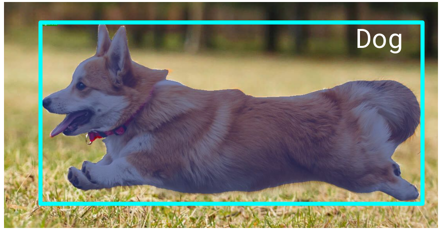
❌ Need manual input
✅ Integrated in labeling tools 🔥

object

Per-SAM:

Mask for similar objects when applied on other images

Grounded-SAM:



Bboxes and masks of the objects prompted

From a foundation model to open world models 💚

Models' labeling power

SAM → Per-SAM → Grounded-SAM

Input

SAM:

Image + click or bounding-box

Per-SAM:

Reference image + mask

Grounded-SAM:

"Dog"

Image + prompt

Output

SAM:

❌ Need manual input
✅ Integrated in labeling tools 🔥

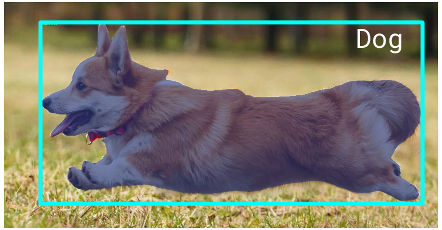
object

Per-SAM:

Didn't work well for my needs ❌

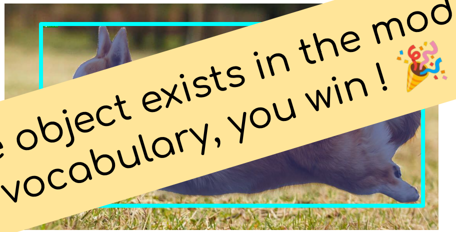
Mask for similar objects when applied on other images

Grounded-SAM:



Bboxes and masks of the objects prompted

From a foundation model to open world models 💚

Models' labeling power	SAM	Per-SAM	Grounded-SAM
Input	 Image + click or bounding-box	 Reference image + mask	 "Dog" Image + prompt
Output	 …ed object	 Mask for similar objects when applied on other images	 Bboxes and masks of the objects prompted

❌ Need manual input
✅ Integrated in labeling tools 🔥

Didn't work well for my needs ❌

If the object exists in the model's vocabulary, you win ! 🎉

From open world models to Vision Language Models

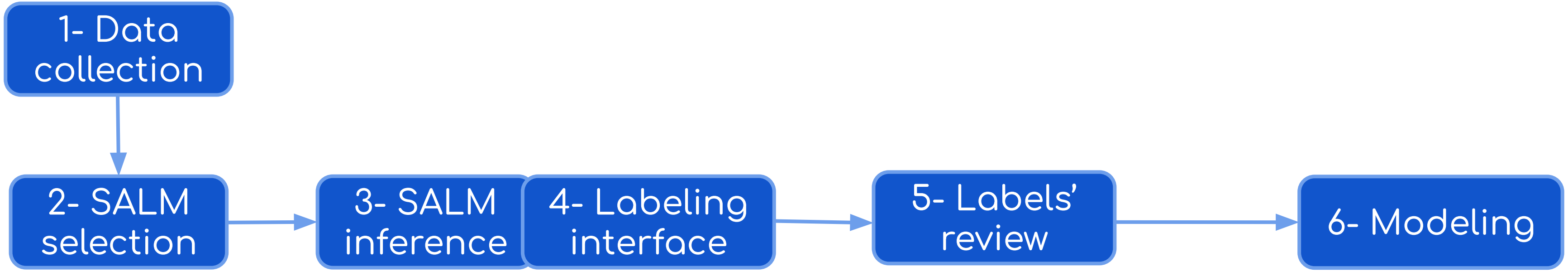
Too many **open-source** options to cite them all:

- Open world models (open-set models)
- Zero-shot models
- Large Multimodal Models
- ...

They are all great candidates to help semi-automate image labeling !

Proposed semi-automated image labeling and model training workflow

SALM = Semi-Automated Labeling Model, eg: Grounding DINO, Grounded-SAM, PaliGemma, Molmo ...



- Test different SALMs
- Test many prompts
- On many samples

Scenario 1:
- DIY inference
- Import images and metadata in the labeling interface

Scenario 2:
- Import images
- Inference using the labeling interface's backend

→ Depending on the SALM and the use case

- Visual inspection of the SALM results
- Use FiftyOne
- Bbox or mask adjustment
- Category adjustment
- Adding missed objects

Build a specialized model
- Use a relevant architecture
- ... that satisfies production constraints
- Apply transfer learning or fine tuning

Takeaways

To make image labeling more efficient:

- Check if you have metadata that can help
- Use a labeling interface and **review** the labels
- Select your modeling strategy wisely
- Take advantage of open-set models and VLMs to semi-automate the process and use this data to train a specialized model

Words by one of my clients CALi in the retail industry:

" ~38% of time was saved compared to a manual labeling process"

Thank you for your attention !

Any questions ?

Follow me on