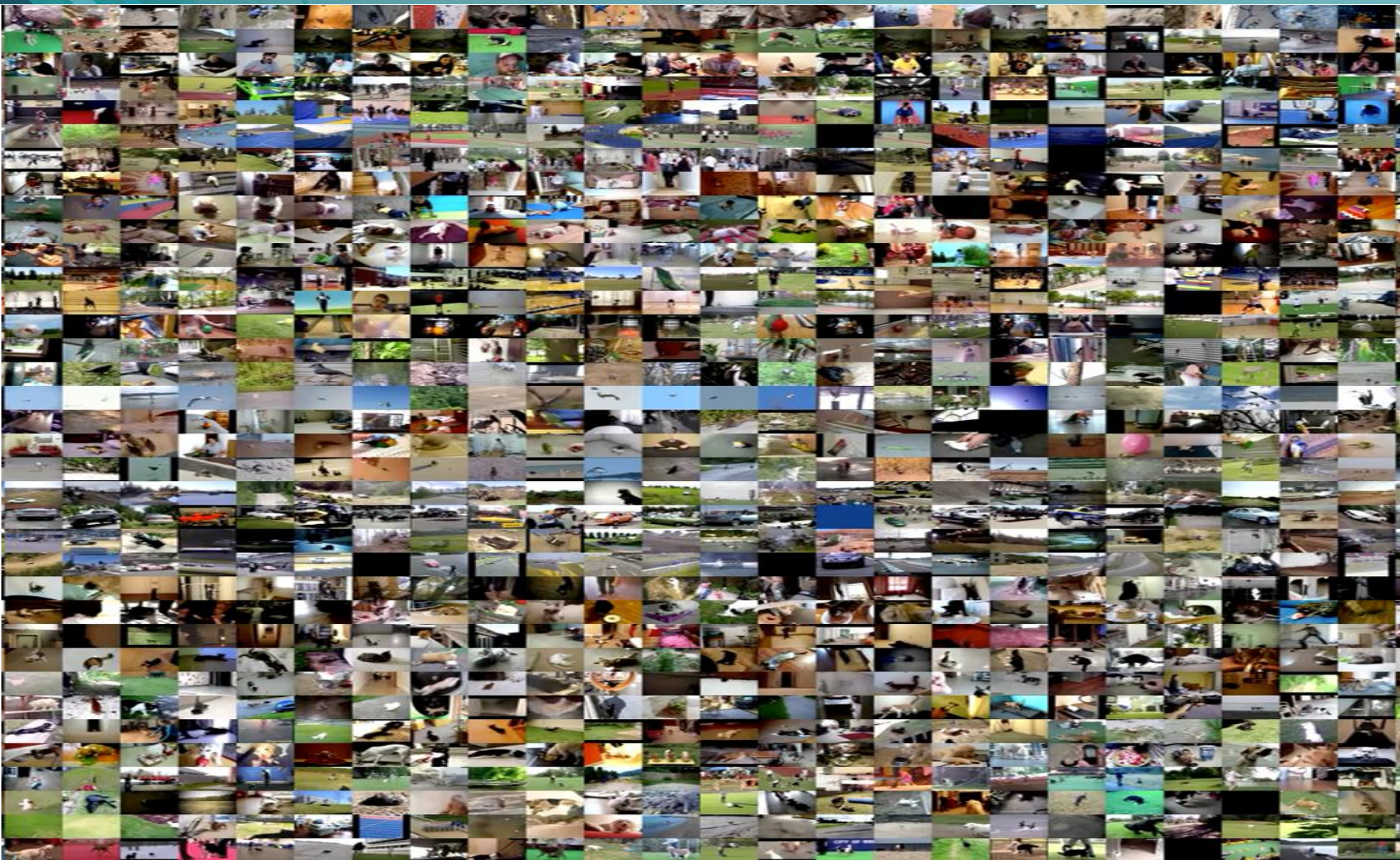




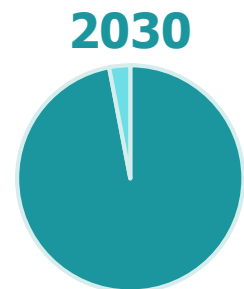
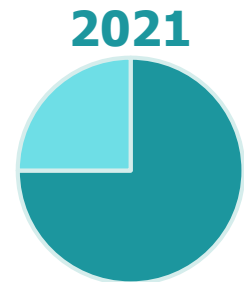
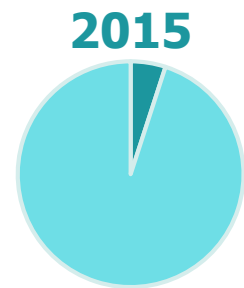
Creating Better Datasets For Training More Robust Models in FiftyOne

Jason Corso, PhD
CEO
Voxel51
<https://fiftyone.ai>

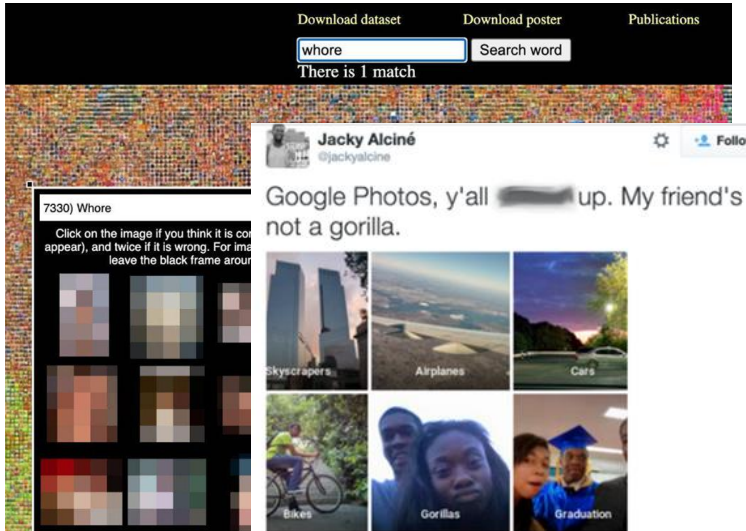
Data Eats Models for Lunch



 **Model work**
 **Data work**



Poor Data? Big Problems



MIT's 80 Million Tiny Images Dataset has ["Significant Ethical Issues"](#)
Google Photos Racial Bias ([WNY Studios](#))

Model Bias Issues



Two killed in Tesla Autopilot Crash with Parked Vehicle ([NYT 2021/08/17](#))

Lethal Physical Danger



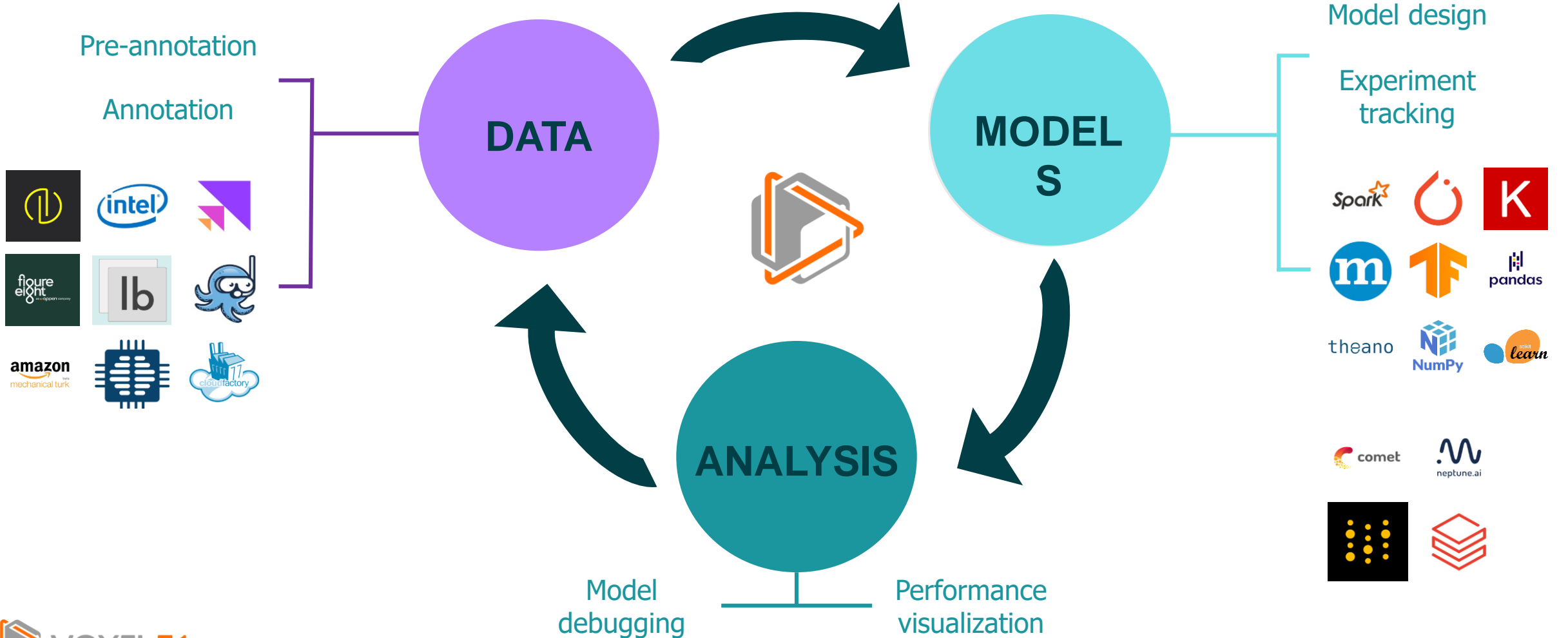
Even Google's Open Images dataset has quality problems ([blog post](#))

30% Reduction in Model Performance

ML Engineers Need Better Tools



To Bring Better Models to Production

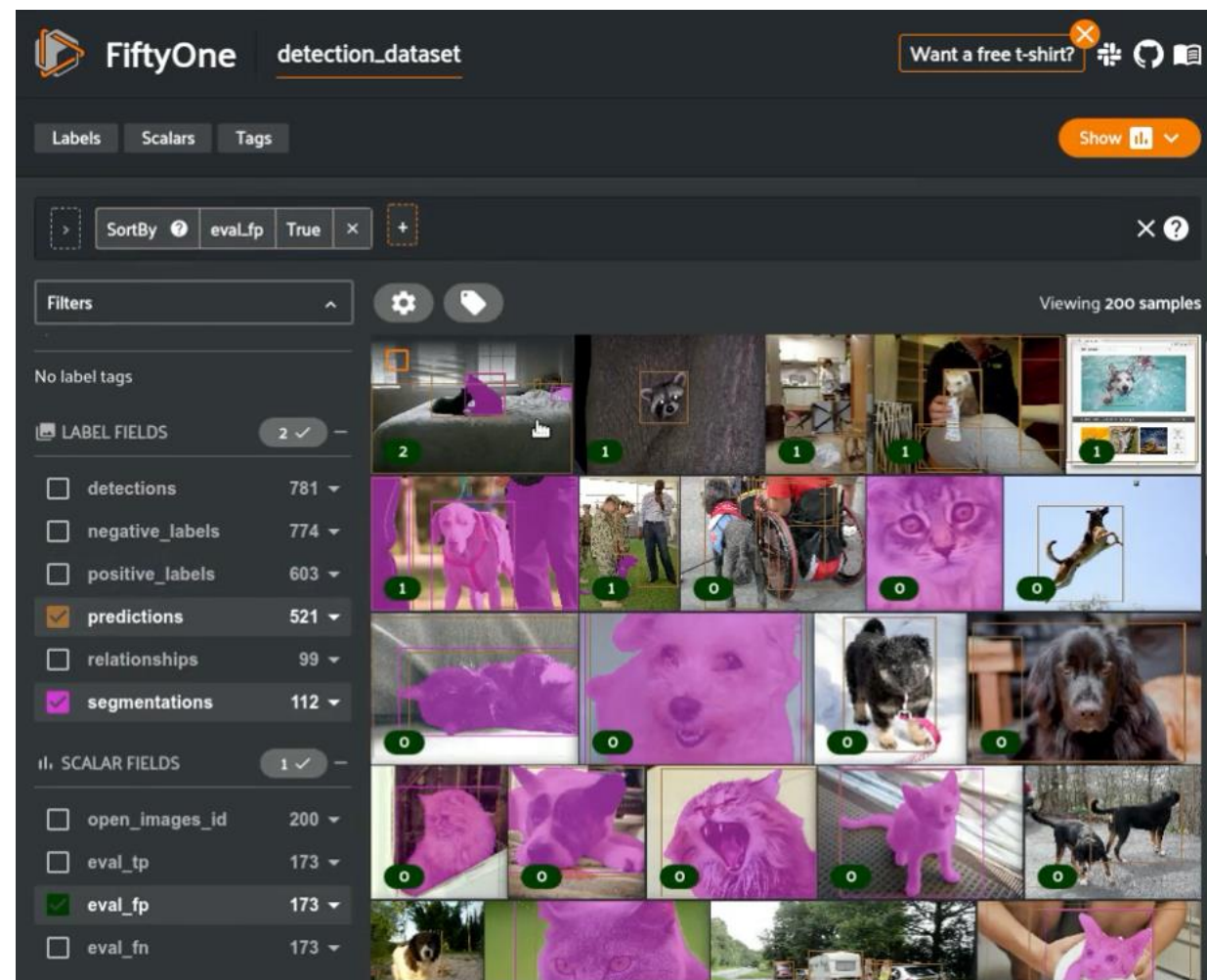


The only open-core data-centric ML solution

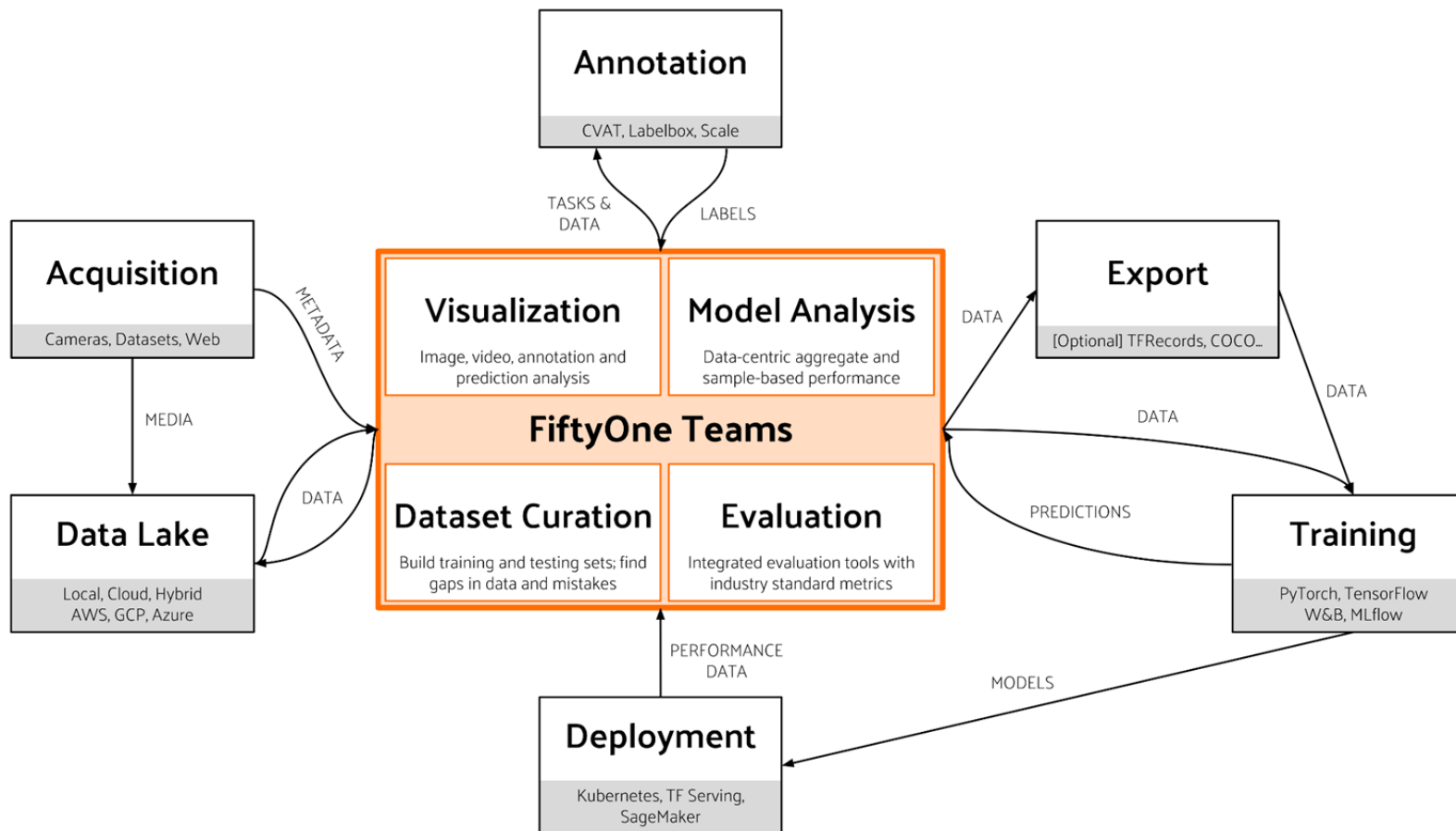
- Curate, visualize and analyze datasets
- Streamline annotation workflows
- Find and fix labeling mistakes
- Identify and correct model failures
- And dozens more workflows...



FiftyOne has helped us rapidly test hypotheses, gain insights about our data, and understand both quantitatively and qualitatively where our models fall short.
– Chris H, Staff Embedded Data Scientist at Vivint



How FiftyOne Fits



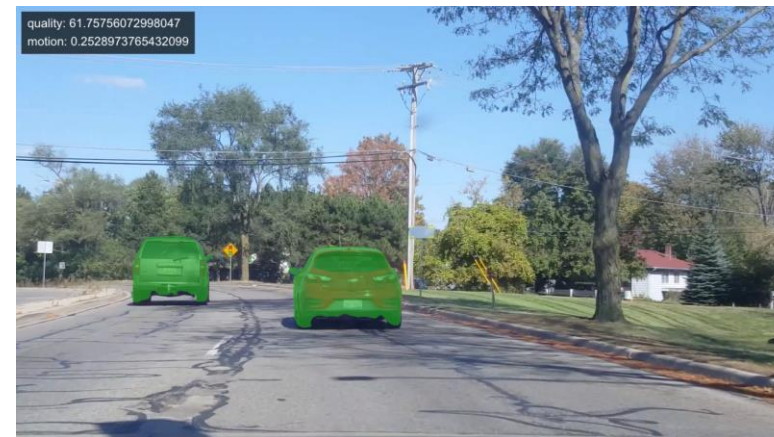
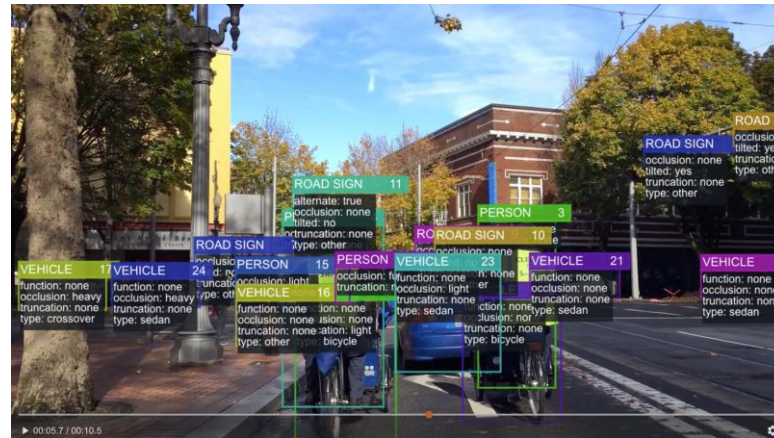
FiftyOne is the single source of truth for ML data on our team

– Mohammed A, Bosch

Supports All Popular Computer Vision Tasks



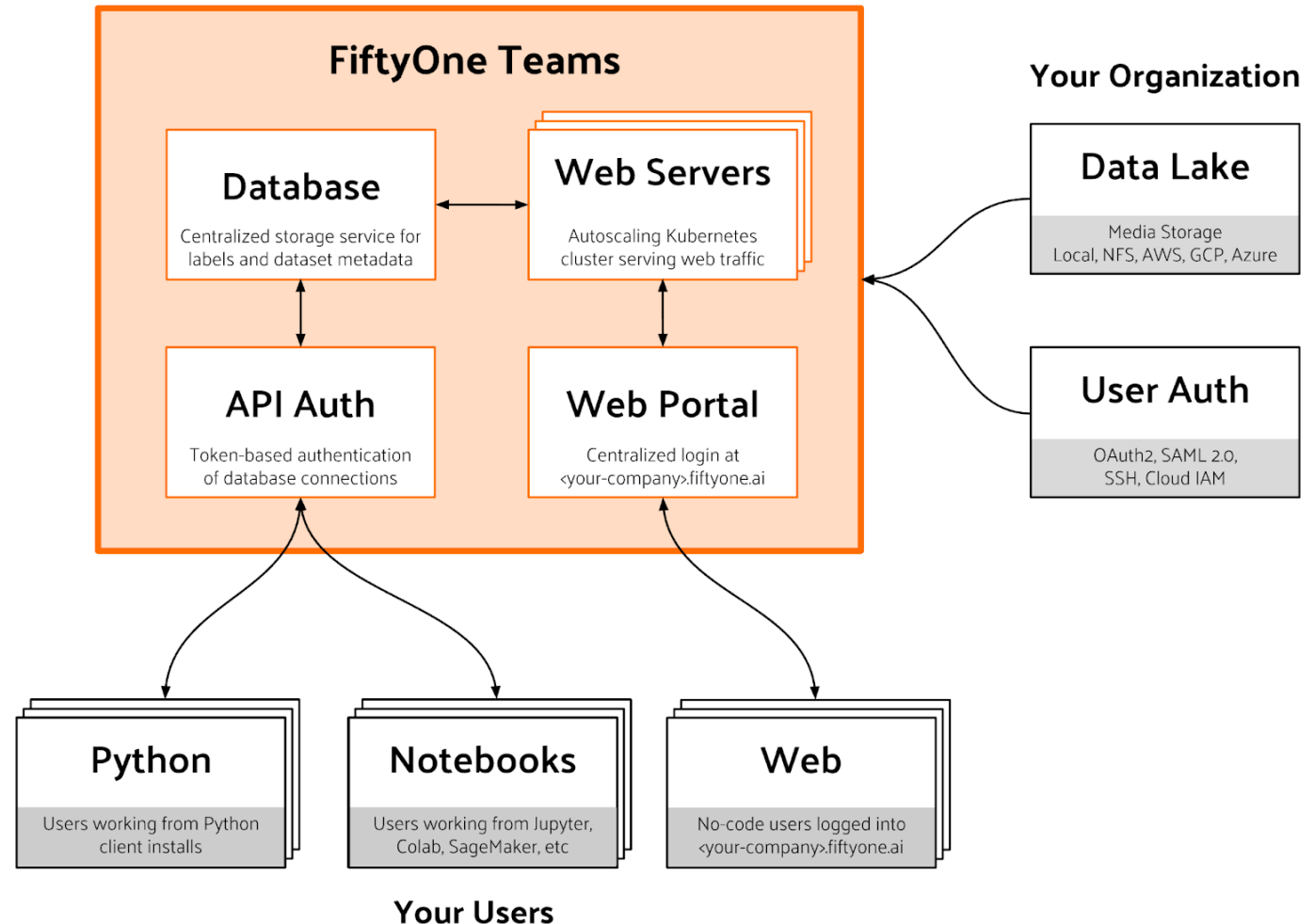
- Classification
- Detection
- Instance segmentation
- Semantic segmentation
- Polygons and polylines
- Keypoints
- **Image and video data**



Data Model and System Layout Brief



- FiftyOne has two key data *types*
 - Dataset
 - View
- Open-source → individuals
- **FiftyOne Teams → groups of users**
 - Same data model
 - Adds “professional” features
 - Shared, cloud-backed data
 - Web portal
 - Versioning



Intuitive and Extensible API



FiftyOne integrates into existing workflows with a few lines of code

- Local, remote, and cloud data
- Handoff between code and App
- Core library is **open source**
- **Integrates with annotation** tools like CVAT, Scale, and Labelbox
- **Integrates with model training** loops in PyTorch, TensorFlow, etc.
- **Integrates with experiment tracking** tools like MLflow and W&B

The screenshot shows a web browser displaying the FiftyOne documentation page for 'Dataset Views'. The page is titled 'Tips & tricks' and contains three sections with code examples:

Chaining view stages

View stages can be chained together to perform complex operations:

```
1 from fiftyone import ViewField as F
2
3 # Extract the first 5 samples with the "validation" tag, alphabetically by
4 # filepath, whose images are >= 48 KB
5 complex_view = (
6     dataset
7     .match_tags("validation")
8     .exists("metadata")
9     .match(F("metadata.size_bytes") >= 48 * 1024) # >= 48 KB
10    .sort_by("filepath")
11    .limit(5)
12 )
```

Filtering detections by area

Need to filter your detections by bounding box area? Use this [ViewExpression](#) !

```
1 from fiftyone import ViewField as F
2
3 # Bboxes are in [top-left-x, top-left-y, width, height] format
4 bbox_area = F("bounding_box")[2] * F("bounding_box")[3]
5
6 # Only contains boxes whose area is between 5% and 50% of the image
7 medium_boxes_view = dataset.filter_labels(
8     "predictions", (0.05 <= bbox_area) & (bbox_area < 0.5)
9 )
```

FiftyOne stores bounding box coordinates as relative values in [0, 1]. However, you can use the expression below to filter by absolute pixel area:

```
1 from fiftyone import ViewField as F
2
3 dataset.compute_metadata()
4
5 # Computes the area of each bounding box in pixels
6 bbox_area = (
7     F("${metadata.width}") * F("bounding_box")[2] *
8     F("${metadata.height}") * F("bounding_box")[3]
9 )
10
11 # Only contains boxes whose area is between 32^2 and 96^2 pixels
12 medium_boxes_view = dataset.filter_labels(
13     "predictions", (32 ** 2 < bbox_area) & (bbox_area < 96 ** 2)
14 )
```

Removing a batch of samples from a dataset

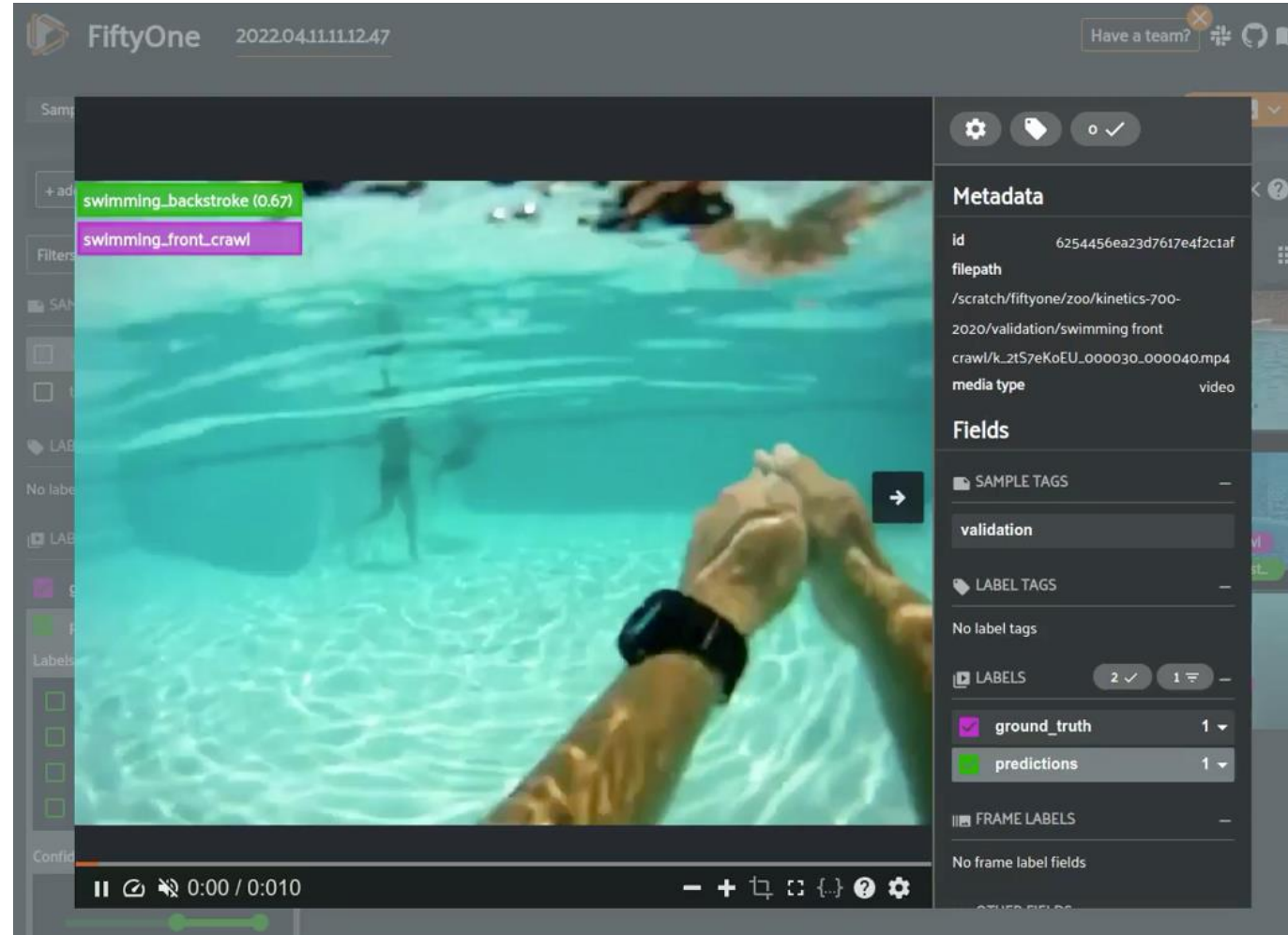
FiftyOne has rich documentation at <https://fiftyone.ai>

FiftyOne Integrations Enhance Workflows



Open-source nature lets FiftyOne extend to meet any ecosystem.

- For example, easily leverage open-source data
- Open Images, COCO, Activity-Net, Kinetics, ...
- Example here visualizes an annotation mistake in Kinetics
- Recent [blog](#) on kinetics



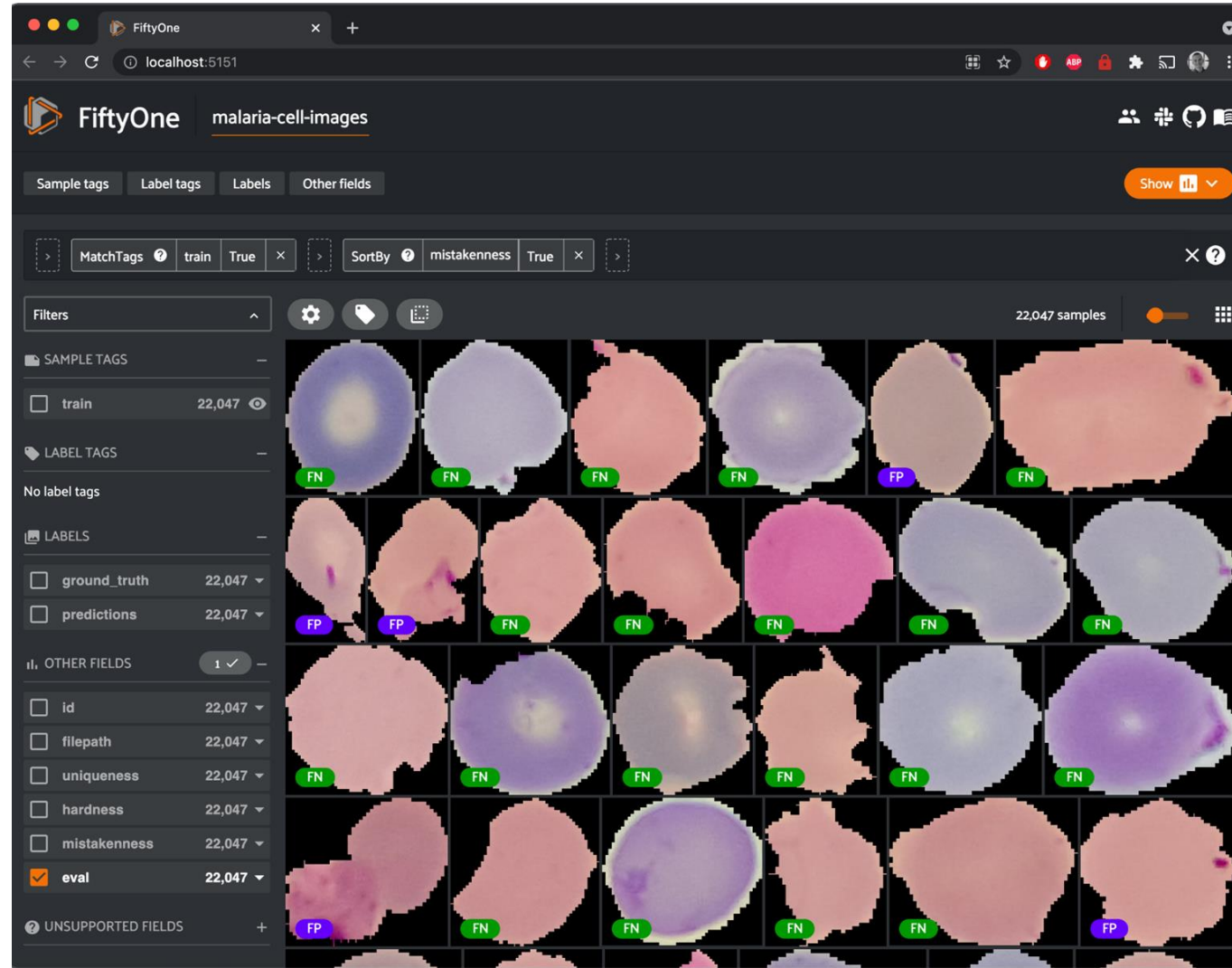
Getting closer to your data means **better datasets** and hence **better models.**

FiftyOne Brain



Automate the data-centric workflows you need to improve your datasets and models

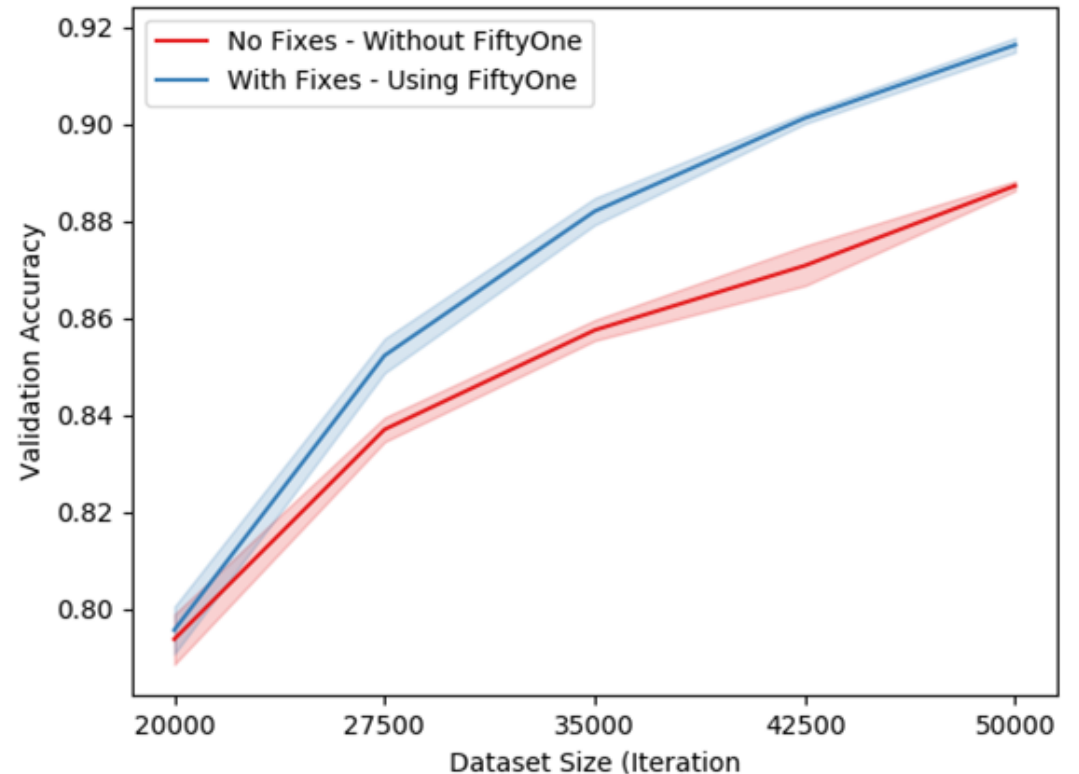
- Identify model failure modes
- Recommend samples for annotation
- Find annotation mistakes
- Embedding-based workflows



Exploring model predictions interactively in the FiftyOne App with best in class model analysis capabilities like false negative and false positive mining

FiftyOne **unlocks higher performance from your models** by identifying and removing the flaws in your data

- Example of using FiftyOne to find and fix annotation mistakes vs random fixes
- Real world scenario of iteratively growing your dataset as you improve your model



Using FiftyOne to find and correct labeling mistakes during model training (“with fixes”) to accelerate training.

Take Home Message



- Data trumps models in production vision systems
- FiftyOne is the leading open-source software for data-centric machine learning workflows
- We offer both free OSS and commercial Teams-based software for augmenting nearly any CV workflow
- **Better Data; Better Models**

Documentation here!

As easy as
pip install fiftyone

<https://fiftyone.ai>

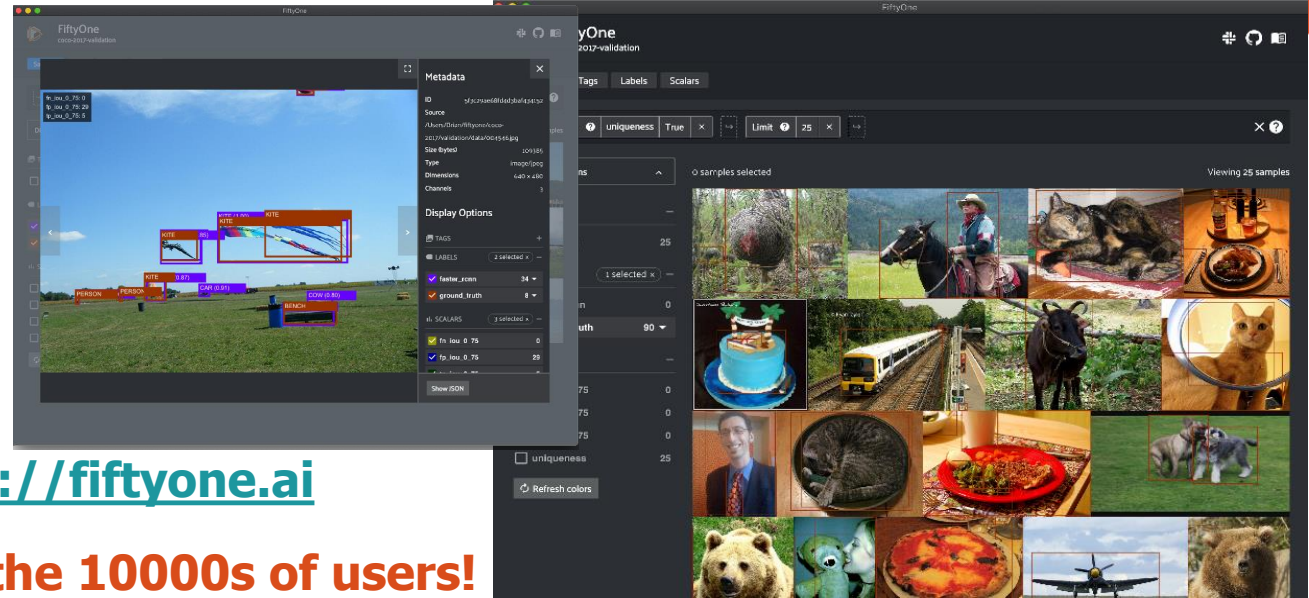
Documentation here!

Next steps

- Like the project?
[Give us a star on GitHub](#)
- Want to get involved?
[Join our Slack community](#)

Light reading

- [Overview blog post 1](#) and [2](#)
- [Installation guide](#)
- [Documentation](#)
- [Tutorials](#)



<https://fiftyone.ai>

Join the 10000s of users!



Thank you!