



# Getting Started with **FiftyOne + Open Source**

*Overview, Installation, Basic Concepts,*

# Housekeeping > About Voxel51 & the Instructor

## Who is Voxel51?

- › The lead maintainers of the open source FiftyOne toolset

## What exactly is a “voxel”?

- › A voxel is to a video what a pixel is to an image. It is literally, a “volume element” in the space-time volume of a video.

## So, what's with the “51”?

- › Good question!



**Daniel Gural**  
ML Engineer & Developer  
Evangelist at Voxel51

# Abridged History of AI/ML

$$T^2 = \frac{4\pi^2}{GM} a^3$$

can be expressed  
as simply

$$T^2 = a^3$$

If expressed in the following units:

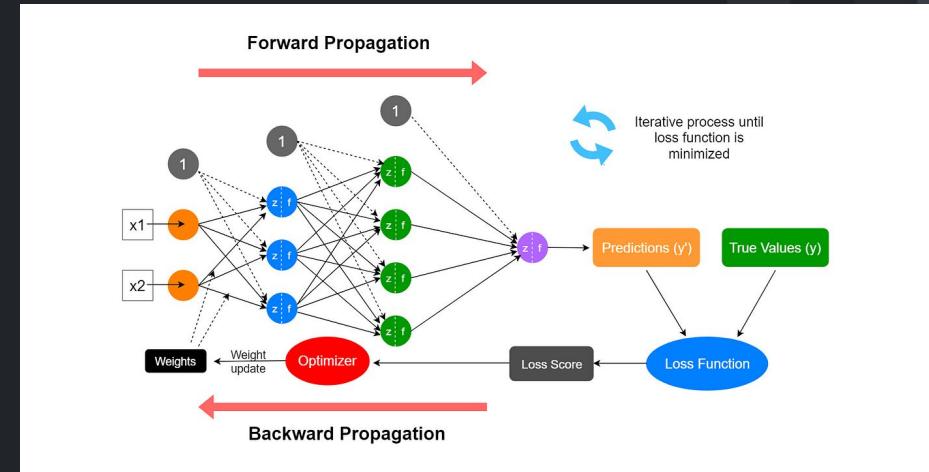
$T$  Earth years

$a$  Astronomical units AU  
( $a = 1$  AU for Earth)

$M$  Solar masses  $M_{\odot}$

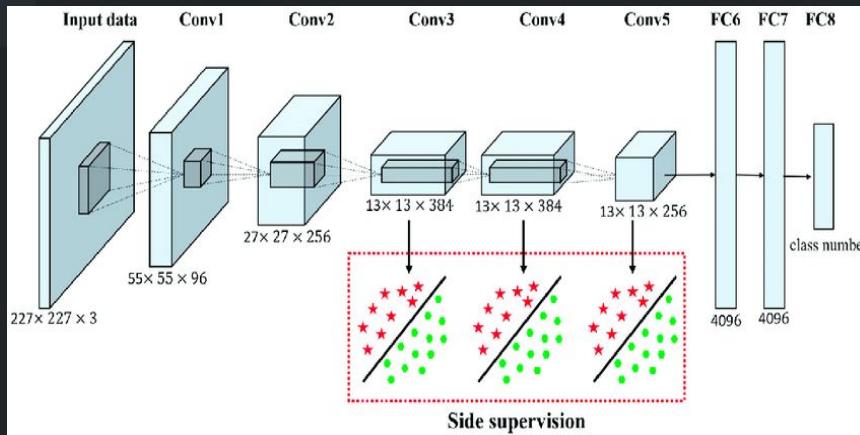
Then  $\frac{4\pi^2}{G} = 1$

From creating formulas  
to ...

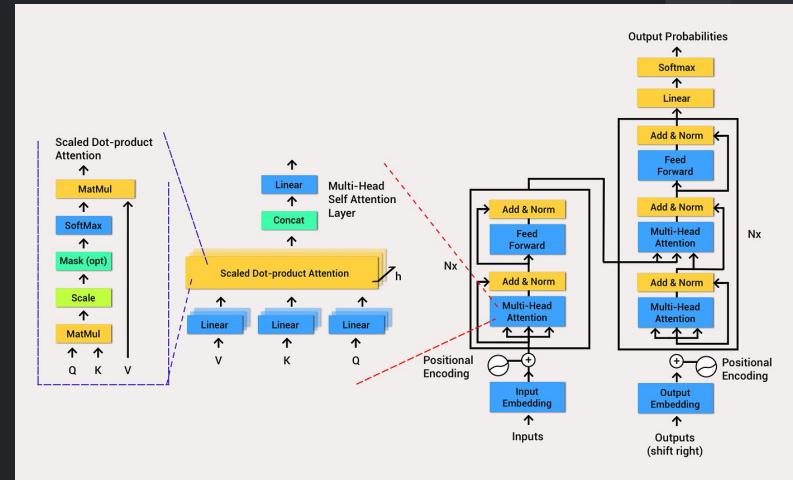


Creating methods to  
predict the formula

# Abridged History of AI/ML

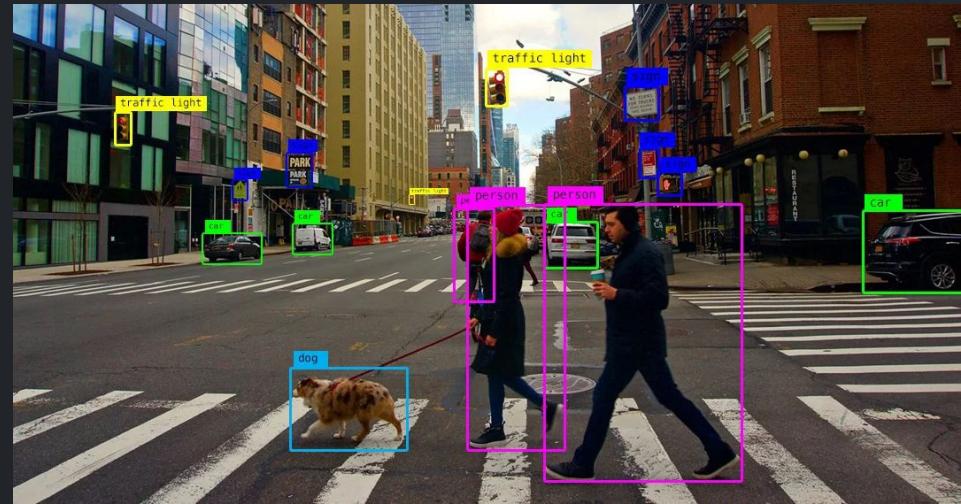
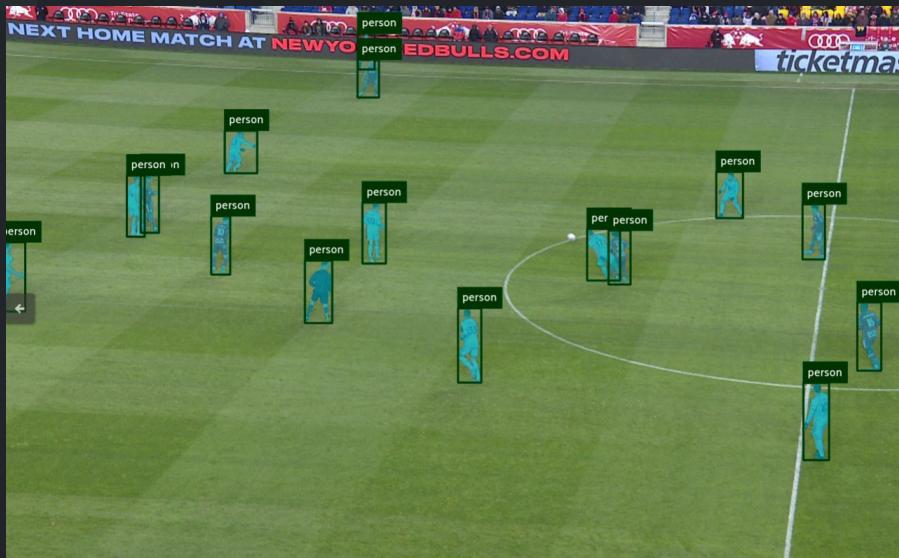


AlexNet (2012)  
62 million



ChatGPT 4 (2023)  
1.76 trillion parameters

# What's Computer Vision?





# The Villain of ML: Bad Data



**Racial Bias Found in a Major Health Care Risk Algorithm**  
Black patients lose out on critical care when systems equate health needs with costs  
By Starre Vartan on October 24, 2019

**Model Bias Issues**



voxel51.com

**Los Angeles Times**  
BUSINESS

11 more crash deaths are linked to automated-tech vehicles

[\(LA Times\)](#)

**Lethal Physical Danger**



**Machine Bias**  
There's software used across the country to predict future criminals. And it's biased against blacks.  
by Julia Angwin, Jeff Larson, Surya Mattu and Lauren Kirchner, ProPublica  
May 23, 2016  
[\(Pro Publica\)](#)

**Reduction in Model Performance**

# Build Better Datasets

# Build Better Datasets



# Teaser

FiftyOne quickstart

+ add stage

X ? Have a Team? ☀️ 🌐 🔍

Samples

200 samples

Unsaved view

FILTER

TAGS

METADATA

LABELS

ground\_truth

predictions

PRIMITIVES

id

filepath

uniqueness

+ ADD GROUP

200 samples

# Teaser

FiftyOne quickstart

+ add stage

X ? Have a Team? ☰

Samples +

200 samples

Unsaved view

FILTER

TAGS

METADATA

LABELS

ground\_truth

predictions

PRIMITIVES

id

filepath

uniqueness

+ ADD GROUP

The interface displays a 5x4 grid of sample images. The images include various subjects such as a bird in a nest, a person riding a horse, a cat sleeping, a meal on a plate, a blue frosted cake, a train on tracks, a goat, an orange cat in a mirror, a man in a suit, a cat's reflection in a bowl, a plate of food, two dogs playing, a large brown bear, a green frog, a pizza, a propeller plane, a grizzly bear, a cat looking at a phone, a bear cub, and a dog wearing a collar.

# Teaser

The image shows a screenshot of the FiftyOne application interface. On the left, there is a grid of sample images from a dataset. Most images feature blue bounding boxes drawn over them, likely indicating detected objects. The sidebar on the right contains a log from the VoxelGPT AI system, which has identified potential algorithms, runs, fields, and label classes based on the query "images with at least two people". The log also shows the Python code used to generate the view, and a message about remembering previous interactions.

FiftyOne voxelgpt

Match ⚡ ... ×

Samples

53 samples

VoxelGPT

Have a Team? 🌐

images with at least two people

Identified potential algorithms: `metadata`

Identified potential runs:

- `metadata` num `metadata`

Identified potential fields: `ground_truth`, `predictions`

Identified potential label classes:

- `ground_truth` field: `person`
- `predictions` field: `person`

Identified potential view stages: `match`, `filter-labels`, `limit`, `exclude`, `exclude-by`

Okay, I'm going to load:

```
python  
view.match(F("ground_truth.detections").filter(F("label") == "per...")
```

I'm remembering your previous view. Any follow-up questions in this session will be posed with respect to it

START OVER

Send a message... ➤

VoxelGPT is in beta and may not understand certain queries. Learn more

# Workshop Agenda

- > Overview
- > Installation & Environments
- > FiftyOne Basic Concepts
- > Open Source
- > Next Steps

# Overview > What Is FiftyOne?

## In a Nutshell

- › Visualize, clean, and curate
- › Find hidden structure
- › Evaluate models
- › *Flexible. Customizable. Connected.*

**An open source tool for building high-quality datasets and computer vision models**



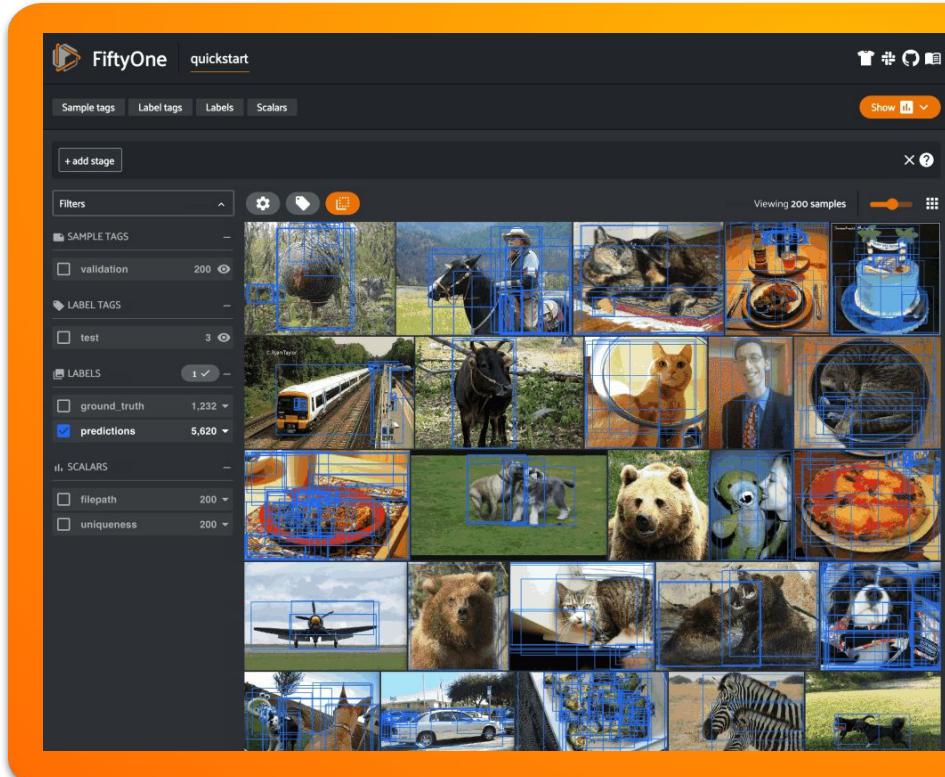
# Overview > Core Capabilities > Curate

- > *Find* : *filter, match, sort, select*
- > *Remove* : duplicates
- > *Add* : tags, metadata, predictions
- > *Correct* : annotation mistakes
- > *Save* : interesting “views”



# Overview > Core Capabilities > Curate

- > **Find** : filter, match, sort, select
- > Remove : duplicates
- > Add : tags, metadata, predictions
- > Correct : annotation mistakes
- > Save : interesting “views”



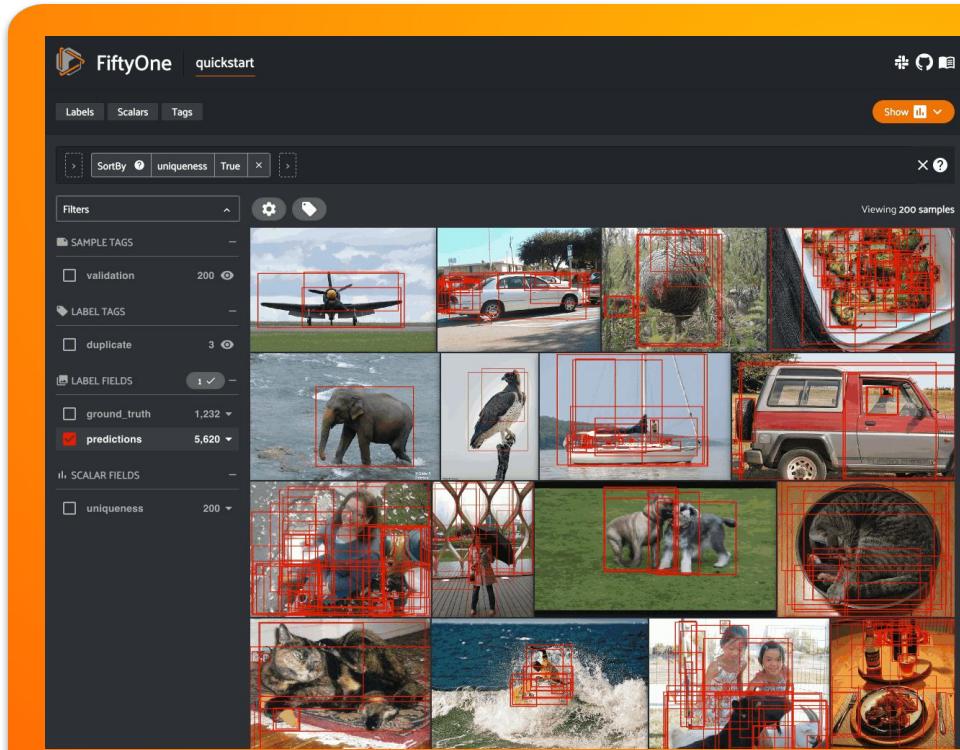
# Overview > Core Capabilities > Curate

- > *Find*: filter, match, sort, select
- > ***Remove : duplicates, spurious labels***
- > *Add*: tags, metadata, predictions
- > *Correct*: annotation mistakes
- > *Save*: interesting “views”



# Overview > Core Capabilities > Curate

- > *Find*: filter, match, sort, select
- > *Remove*: duplicates
- > **Add : tags, metadata, predictions**
- > *Correct*: annotation mistakes
- > *Save*: interesting “views”



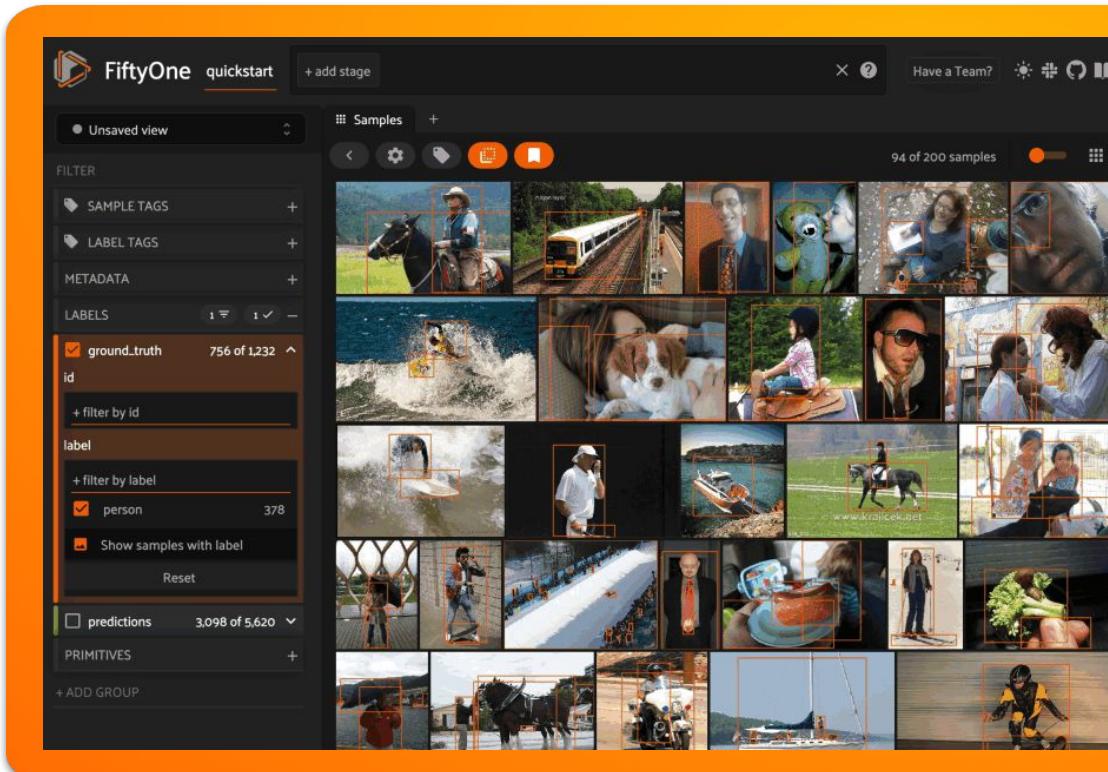
# Overview > Core Capabilities > Curate

- > *Find*: filter, match, sort, select
- > *Remove*: duplicates
- > *Add*: tags, metadata, predictions
- > ***Correct : annotation mistakes***
- > *Save*: interesting “views”

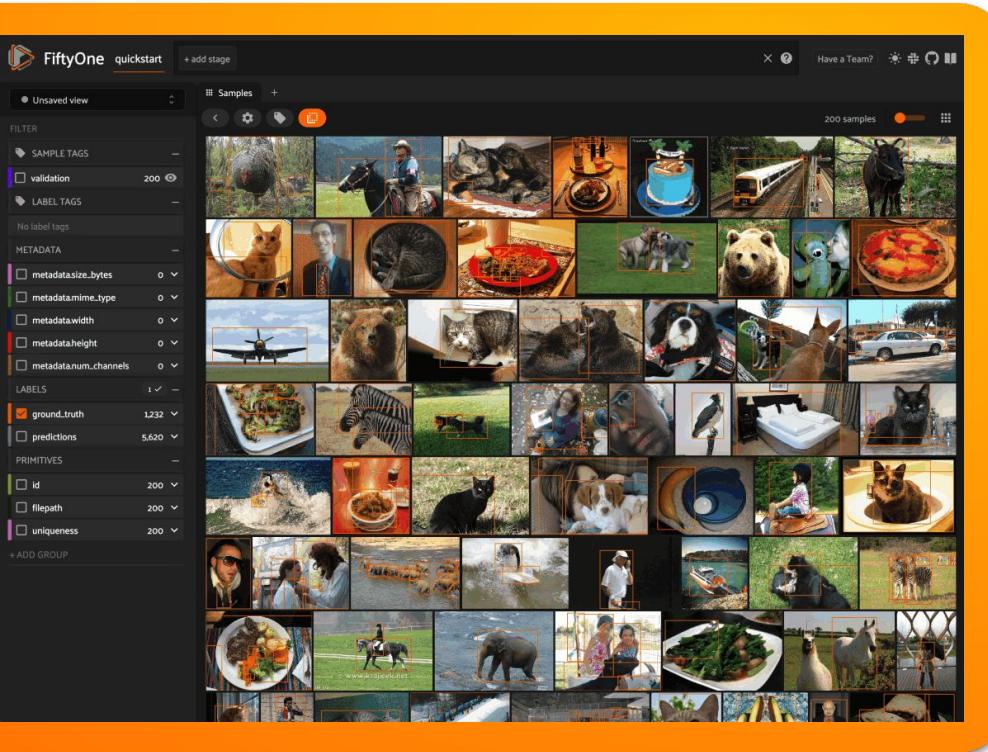


# Overview > Core Capabilities > Curate

- › *Find*: filter, match, sort, select
- › *Remove*: duplicates
- › *Add*: tags, metadata, predictions
- › *Correct*: annotation mistakes
- › ***Save : interesting “views”***

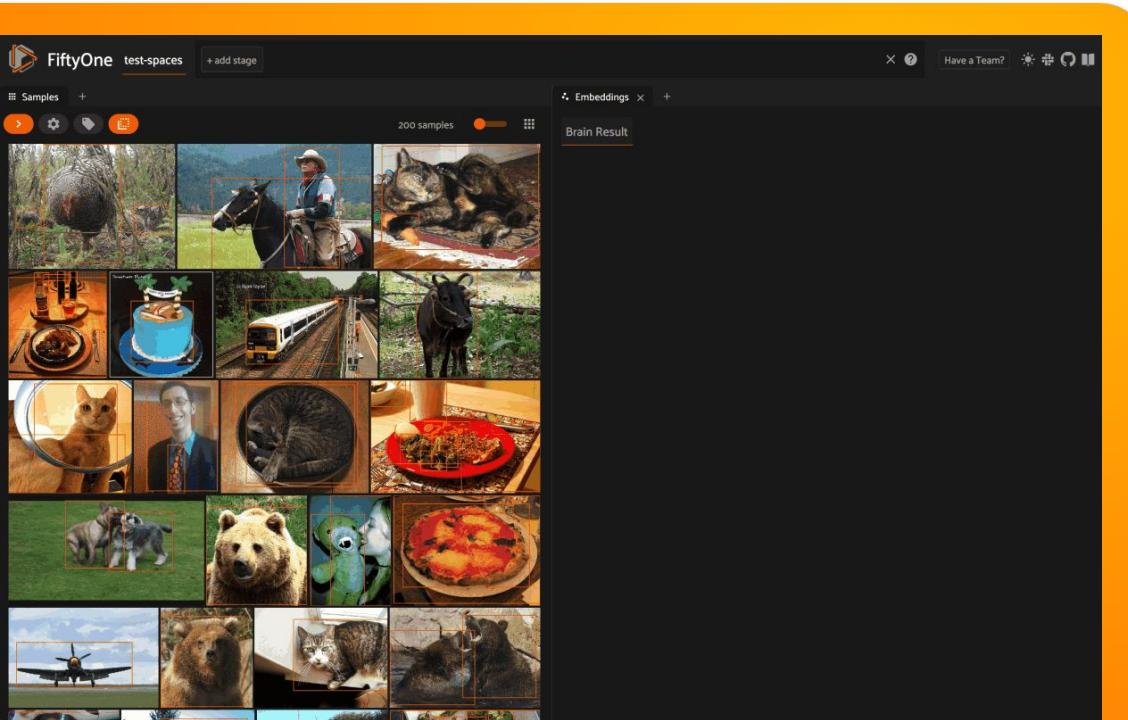


# Overview > Core Capabilities > Understand



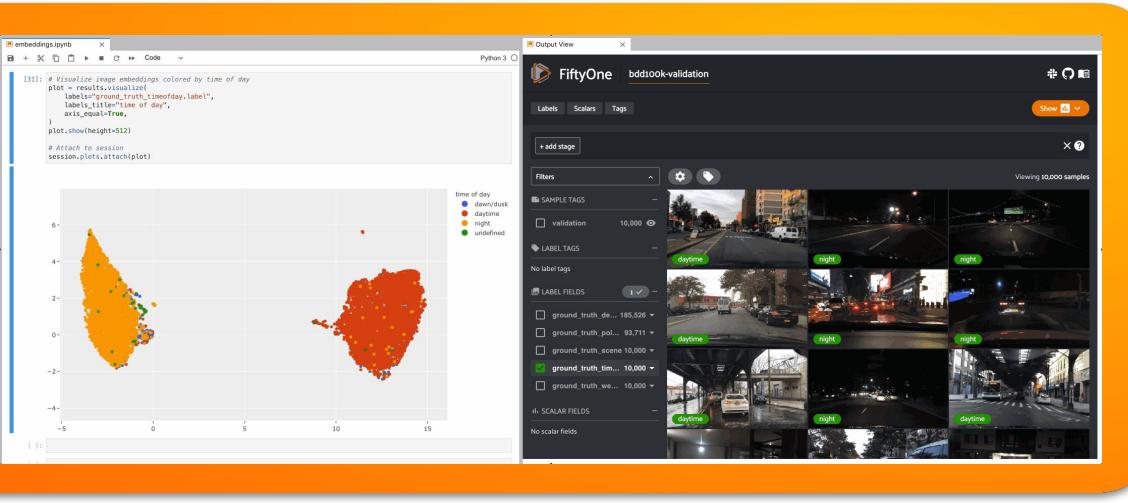
- > **Aggregate statistics**
- > Embeddings
- > Interactive visualization

# Overview > Core Capabilities > Understand



- › Aggregate statistics
- › *Embeddings*
- › Interactive visualization

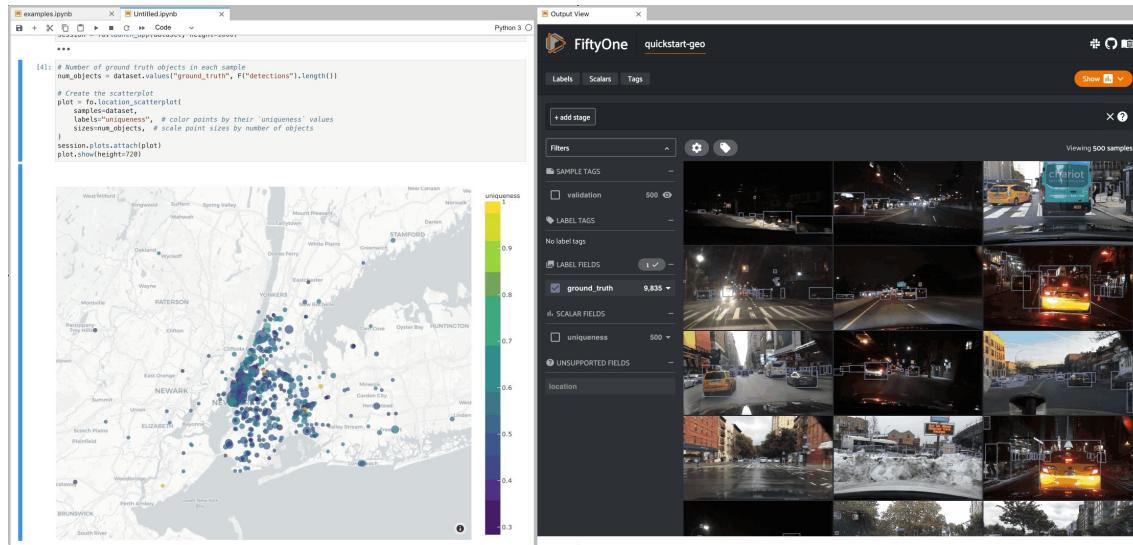
# Overview > Core Capabilities > Understand



- > Aggregate statistics
- > *Embeddings*
- > Interactive visualization

# Overview > Core Capabilities > Understand

- > Aggregate statistics
- > Embeddings
- > *Interactive visualization*



# Overview > Core Principles > Flexible

## Datasets

- › Format conversion
- › Merging, adding to, cloning
- › Import/export
- › Load from Dataset Zoo
- › Define custom schema

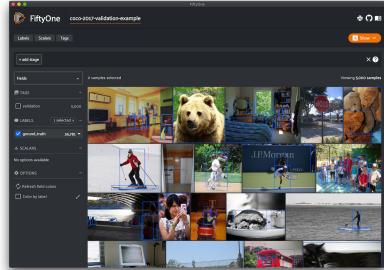
## Models

- › BYO or load from Model Zoo
- › PyTorch, TensorFlow, or Sklearn

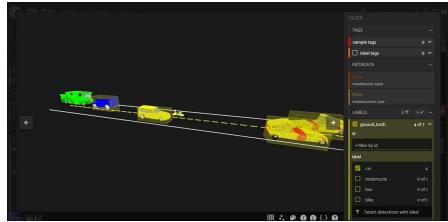
Dataset format	Import Supported?	Export Supported?	Conversion Supported?
<a href="#">ImageDirectory</a>	✓	✓	✓
<a href="#">VideoDirectory</a>	✓	✓	✓
<a href="#">FiftyOneImageClassificationDataset</a>	✓	✓	✓
<a href="#">ImageClassificationDirectoryTree</a>	✓	✓	✓
<a href="#">TFImageClassificationDataset</a>	✓	✓	✓
<a href="#">FiftyOneImageDetectionDataset</a>	✓	✓	✓
<a href="#">COCODetectionDataset</a>	✓	✓	✓
<a href="#">VOCDetectionDataset</a>	✓	✓	✓
<a href="#">KITTDetectionDataset</a>	✓	✓	✓
<a href="#">YOLODataset</a>	✓	✓	✓
<a href="#">TFOBJECTDetectionDataset</a>	✓	✓	✓
<a href="#">CVATImageDataset</a>	✓	✓	✓
<a href="#">CVATVideoDataset</a>	✓	✓	✓
<a href="#">FiftyOneImageLabelsDataset</a>	✓	✓	✓



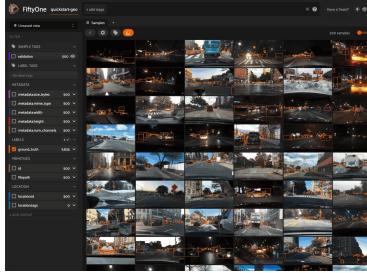
# Overview > Core Principles > Flexible > Media Types



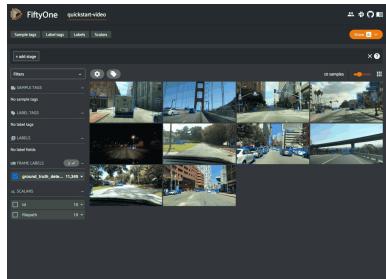
Image



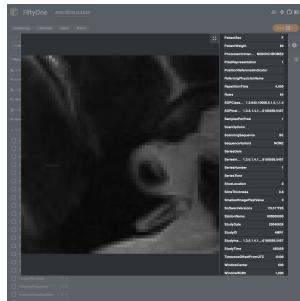
Point cloud



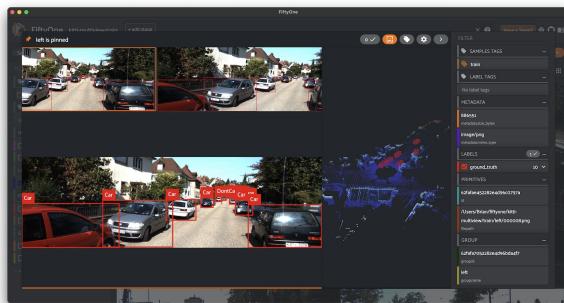
Geo



Video



DICOM

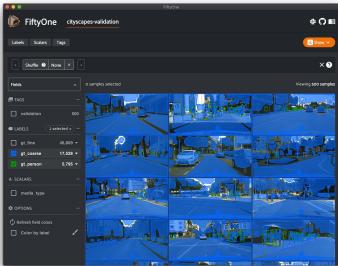


Groups

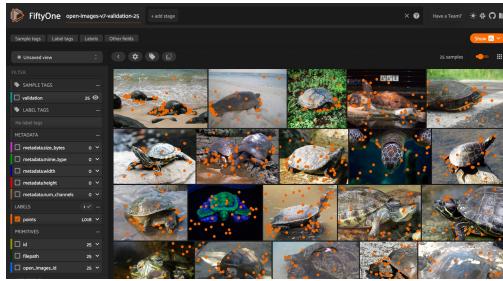
# Overview > Core Principles > Flexible > Labels



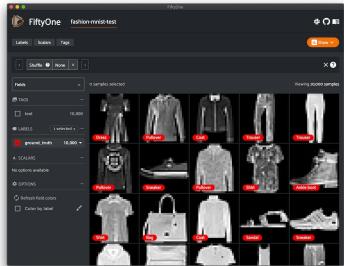
Detection



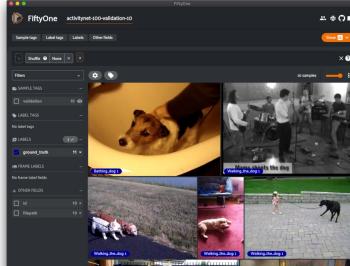
Masks



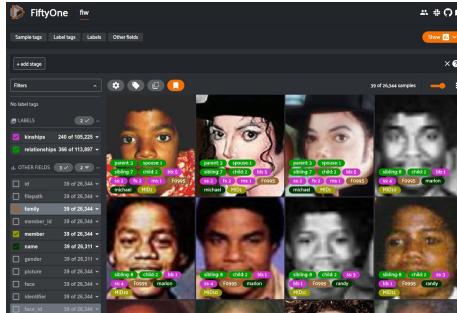
Keypoint



Classification

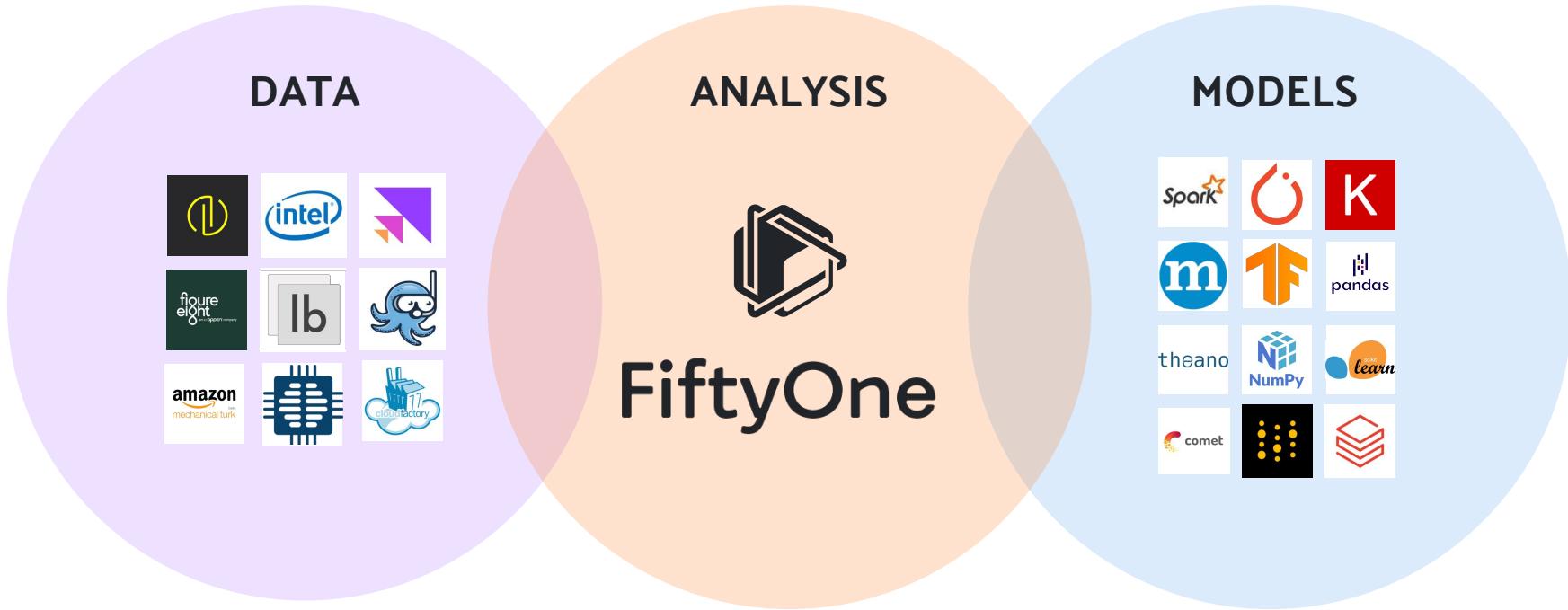


Action recognition

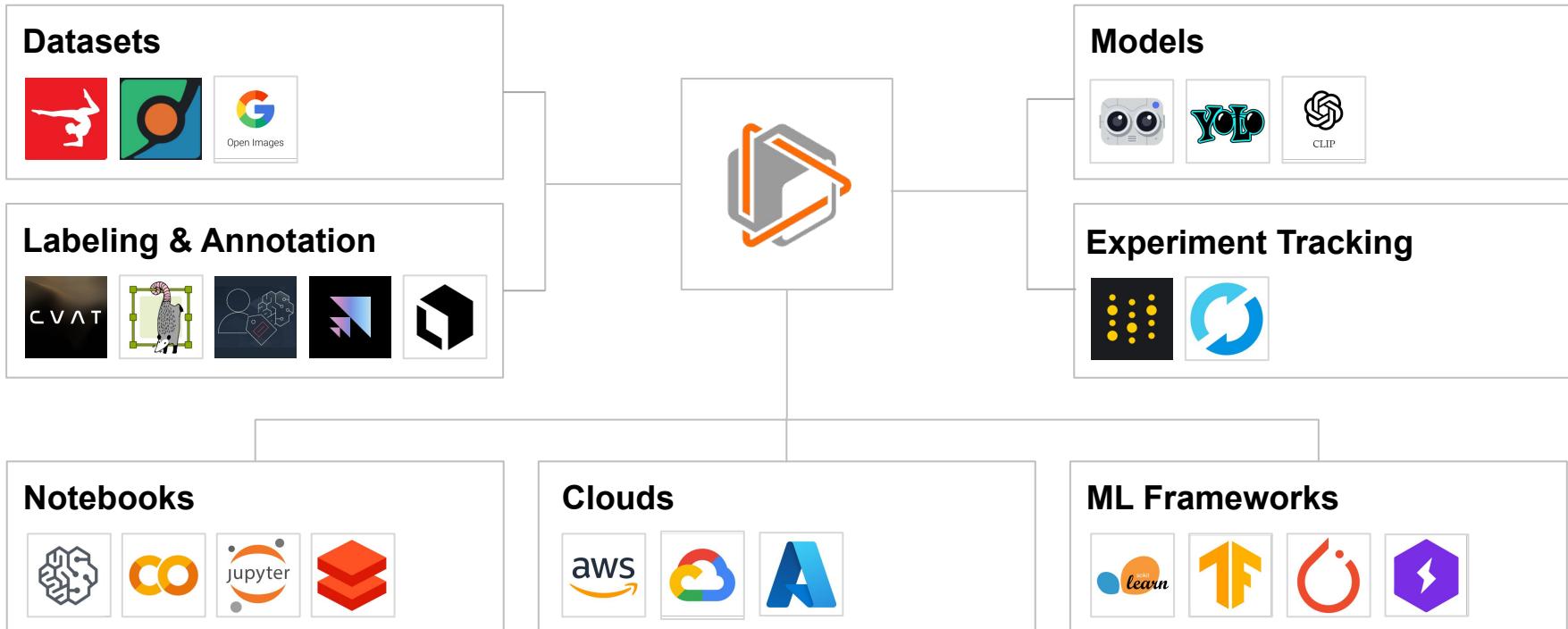


Relationships and more...

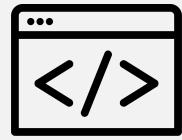
# Where FiftyOne Fits



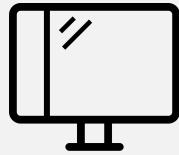
# Overview > Core Principles > Connected



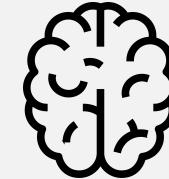
# Overview > FiftyOne



FiftyOne  
Library



FiftyOne  
App



FiftyOne  
Brain

# Workshop Agenda

- > Overview
- > **Installation & Environments**
- > FiftyOne Basic Concepts
- > Open Source
- > Next Steps

# Installation > Prerequisites

## Python

- › Versions 3.7-3.10 currently supported
- › Verify you have Python installed
- › Identify your version of Python

```
$ python --version  
Python 2.7.17
```

```
$ python3 --version  
Python 3.8.9
```

## Virtual or Conda Environment (Recommended)

- › Use `venv` or `conda` to run FiftyOne in an isolated environment to avoid dependency versioning conflicts



# Installation > venv and conda

## Virtual environment

- › Navigate to desired directory
- › Create and activate environment

```
$ cd path/to/dir  
$ python3 -m venv env  
$ . env/bin/activate
```

## Conda environment

- › Download and install Anaconda
- › Create and activate environment

```
$ conda create --name myenv  
$ conda activate myenv
```

## Install FiftyOne

```
$ pip3 install fiftyone
```



# Installation > venv and conda

## Virtual environment

- › Navigate to desired directory
- › Create and activate environment

```
$ cd path/to/dir  
$ python3 -m venv env  
$ . env/bin/activate
```

## Conda environment

- › Download and install Anaconda
- › Create and activate environment

```
$ conda create --name myenv  
$ conda activate myenv
```

## Install FiftyOne WITH DESKTOP APP

```
$ pip3 install "fiftyone[desktop]"
```



# Installation > Import FiftyOne

```
$ python  
  
->>> import fiftyone as fo  
->>> print(fo)  
<module 'fiftyone' from '/Users/...'>
```

# Installation > Extras

## Notebooks

*Jupyter*

## Models

*Torch*

*Torchvision*

*TensorFlow*

## Videos

*FFmpeg*

## Point clouds

*Open3D*

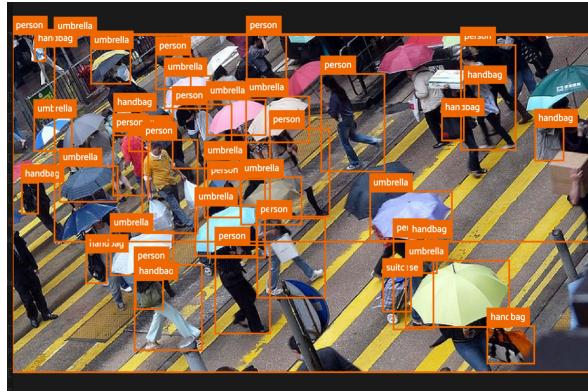


# Workshop Agenda

- > Overview
- > Installation & Environments
- > **FiftyOne Basic Concepts**
- > Open Source
- > Next Steps

# Tabular Data vs Unstructured Data

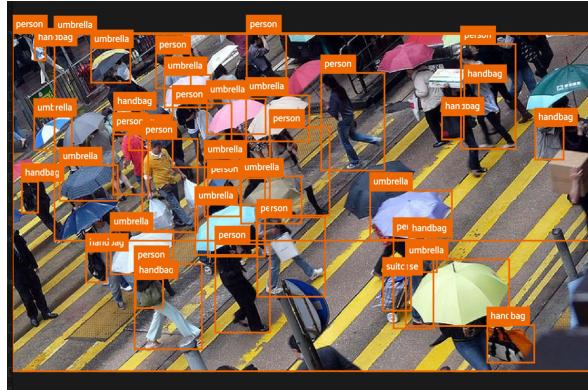
## Computer Vision Data



	Deto	Det1	Det2	Det4	...
Sample0	Bear				
Sample1	Umbrella	Person	Person	Handbag	

# Tabular Data vs Unstructured Data

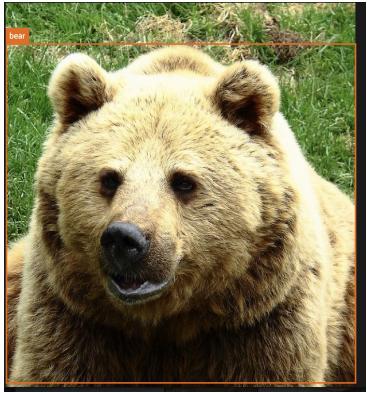
## Computer Vision Data



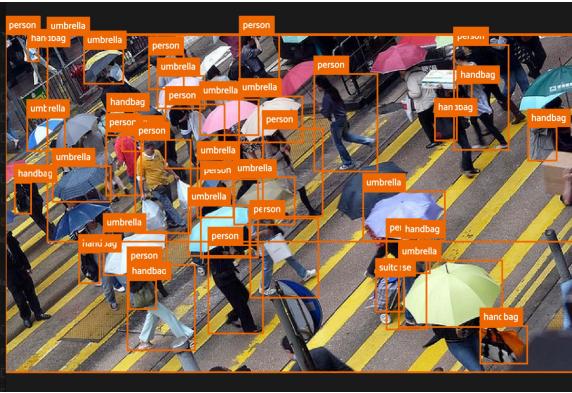
	Deto	Det1	Det2	Det4	...
Sample0	Bear	X	X	X	
Sample1	Umbrella	Person	Person	Handbag	

# Tabular Data vs Unstructured Data

## Computer Vision Data



```
[{"detection": {  
    "id": "5f452460ef00e6374aac215e",  
    "attributes": {},  
    "bbox_2d": [],  
    "label": "bear",  
    "bounding_box": [  
        0.02440273037542662,  
        0.107515625,  
        0.9975597269624574,  
        0.8811562500000001,  
    ],  
    "mask": None,  
    "confidence": None,  
    "index": None,  
    "area": 275705.8110500001,  
    "iscrowd": 0.0,  
}],
```

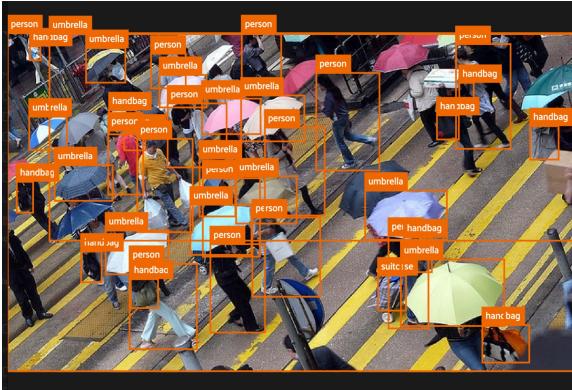


# Tabular Data vs Unstructured Data

## Computer Vision Data



```
[{"id": "5f452460ef00e6374a0c215e",  
 "attributes": {},  
 "bbox_2d": [],  
 "label": "bear",  
 "bounding_box": [  
     0.02440273037542662,  
     0.107515625,  
     0.9975597269624574,  
     0.8811562500000001,  
 ],  
 "mask": None,  
 "confidence": None,  
 "index": None,  
 "area": 275705.8110500001,  
 "iscrowd": 0.0,  
 }]
```



```
[{"id": "5f452460ef00e6374a0c215e",  
 "attributes": {},  
 "bbox_2d": [],  
 "label": "bear",  
 "bounding_box": [  
     0.02440273037542662,  
     0.107515625,  
     0.9975597269624574,  
     0.8811562500000001,  
 ],  
 "mask": None,  
 "confidence": None,  
 "index": None,  
 "area": 275705.8110500001,  
 "iscrowd": 0.0,  
 }]
```

```
[{"id": "5f452460ef00e6374a0c215e",  
 "attributes": {},  
 "bbox_2d": [],  
 "label": "bear",  
 "bounding_box": [  
     0.02440273037542662,  
     0.107515625,  
     0.9975597269624574,  
     0.8811562500000001,  
 ],  
 "mask": None,  
 "confidence": None,  
 "index": None,  
 "area": 275705.8110500001,  
 "iscrowd": 0.0,  
 }]
```

```
[{"id": "5f452460ef00e6374a0c215e",  
 "attributes": {},  
 "bbox_2d": [],  
 "label": "bear",  
 "bounding_box": [  
     0.02440273037542662,  
     0.107515625,  
     0.9975597269624574,  
     0.8811562500000001,  
 ],  
 "mask": None,  
 "confidence": None,  
 "index": None,  
 "area": 275705.8110500001,  
 "iscrowd": 0.0,  
 }]
```

```
[{"id": "5f452460ef00e6374a0c215e",  
 "attributes": {},  
 "bbox_2d": [],  
 "label": "bear",  
 "bounding_box": [  
     0.02440273037542662,  
     0.107515625,  
     0.9975597269624574,  
     0.8811562500000001,  
 ],  
 "mask": None,  
 "confidence": None,  
 "index": None,  
 "area": 275705.8110500001,  
 "iscrowd": 0.0,  
 }]
```

```
[{"id": "5f452460ef00e6374a0c215e",  
 "attributes": {},  
 "bbox_2d": [],  
 "label": "bear",  
 "bounding_box": [  
     0.02440273037542662,  
     0.107515625,  
     0.9975597269624574,  
     0.8811562500000001,  
 ],  
 "mask": None,  
 "confidence": None,  
 "index": None,  
 "area": 275705.8110500001,  
 "iscrowd": 0.0,  
 }]
```

```
[{"id": "5f452460ef00e6374a0c215e",  
 "attributes": {},  
 "bbox_2d": [],  
 "label": "bear",  
 "bounding_box": [  
     0.02440273037542662,  
     0.107515625,  
     0.9975597269624574,  
     0.8811562500000001,  
 ],  
 "mask": None,  
 "confidence": None,  
 "index": None,  
 "area": 275705.8110500001,  
 "iscrowd": 0.0,  
 }]
```

# Tabular Data vs Unstructured Data

## Computer Vision Data

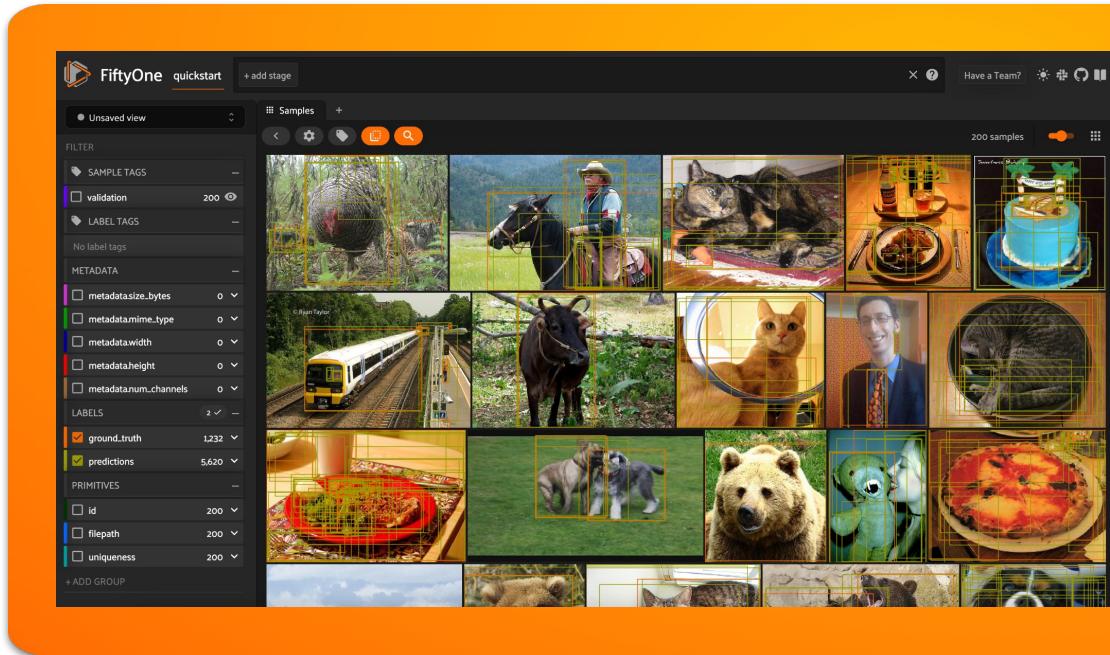


```
[{"Detection": {  
    "id": "5f452d0ef00e6374sec215e",  
    "label": {},  
    "top": 0, // Unstructured data  
    "label": "bear",  
    "bounding_box": [  
        0.00240273037542662,  
        0.10715625,  
        0.997597260624574,  
        0.8815625000000001  
    ],  
    "mask": None,  
    "confidence": None,  
    "index": None,  
    "area": 275799.8110500001,  
    "iscrowd": 0.0,  
    "group": 1  
}]
```



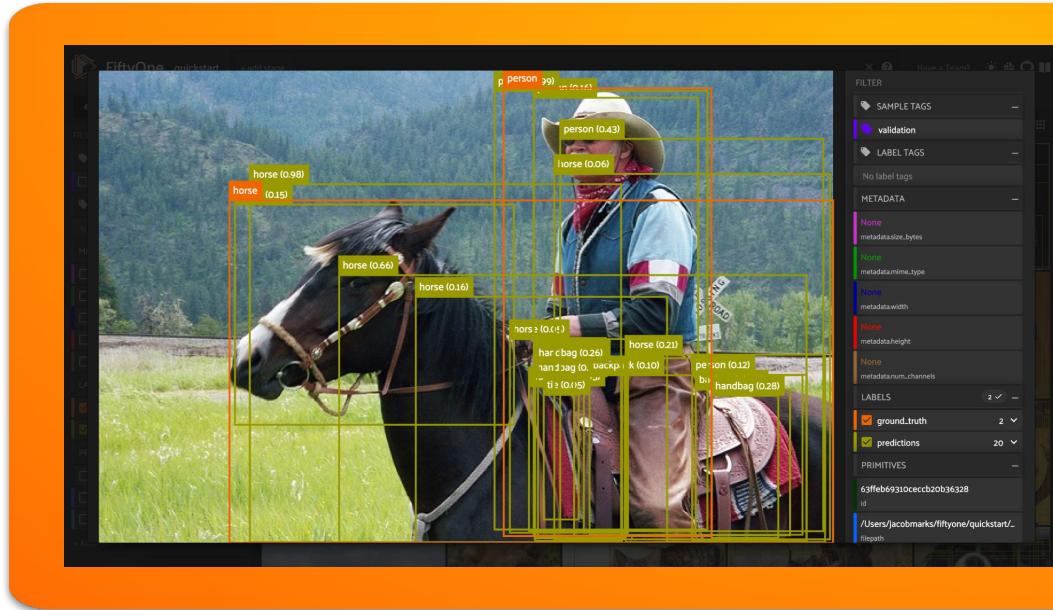
# FiftyOne Basic Concepts > Schema

- The **Dataset** class represents your data and can be manipulated through the Python library and the FiftyOne App



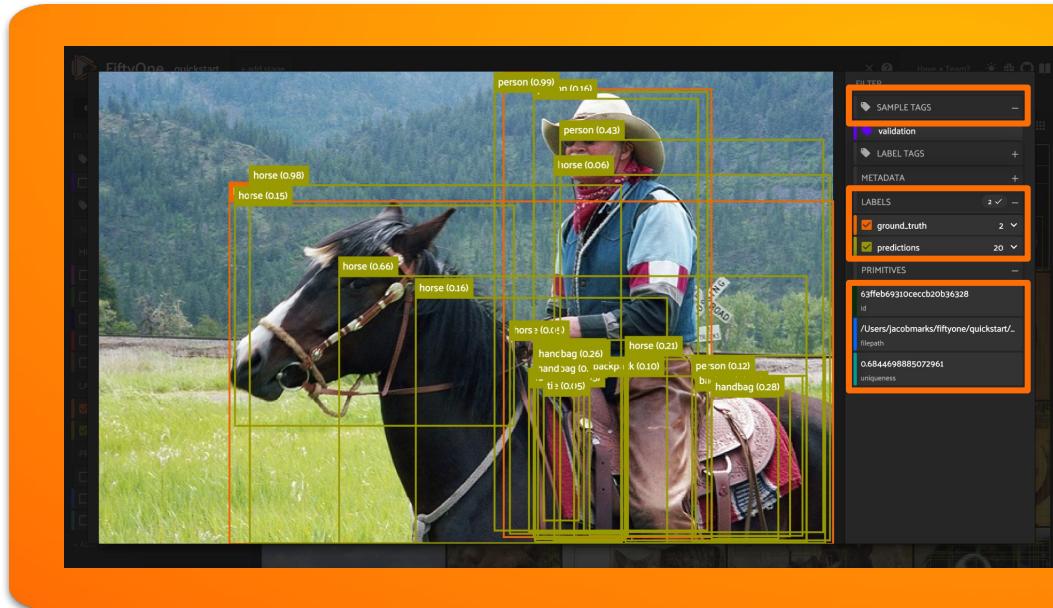
# FiftyOne Basic Concepts > Schema

- **Samples** are the atomic elements of a **Dataset** that store all the information related to a given piece of data, like an image or video



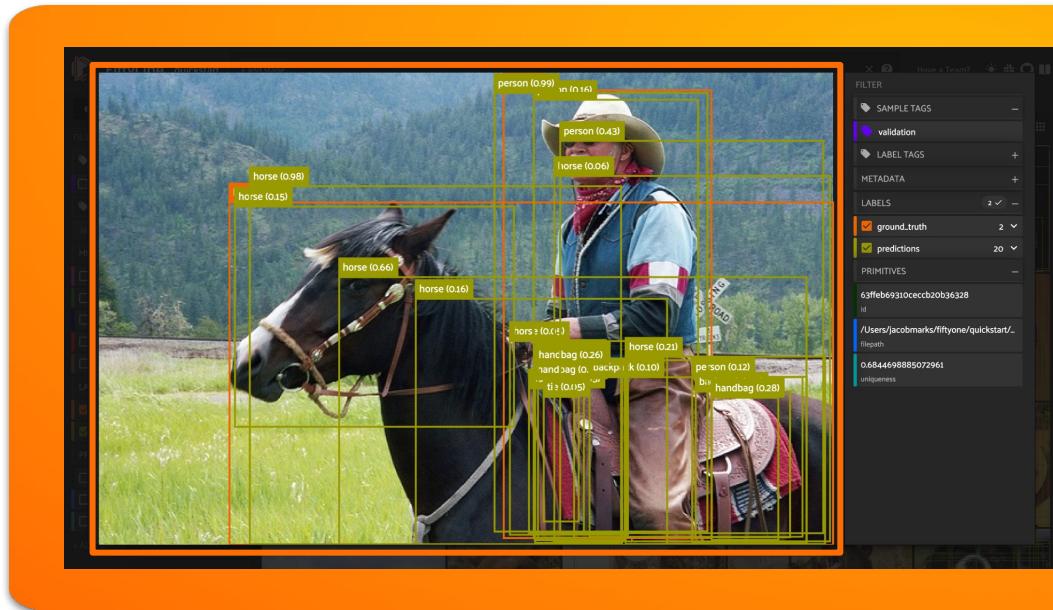
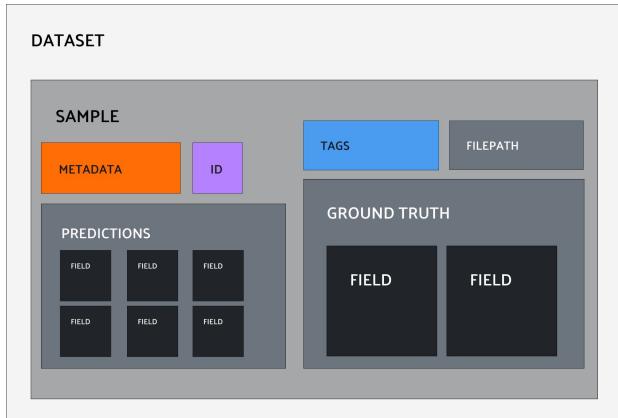
# FiftyOne Basic Concepts > Schema

- A Field stores an attribute. **Filepath**, **sample ID**, and **tags** are fields.



# FiftyOne Basic Concepts > Schema

- A **Field** can contain more complex data, including other fields!



# Workshop Agenda

- > Overview
- > Installation & Environments
- > FiftyOne Basic Concepts
- > **Open Source Projects**
- > Next Steps

# Open Source Projects

## › Why Open Source?



# Open Source Projects

## › Why Open Source?



# Open Source Projects

› New Projects are easy to find

The image displays two screenshots of web interfaces for discovering new open source projects.

**Left Screenshot:** A screenshot of a "Spaces" interface. The header reads "Discover amazing ML apps made by the community!". It features a search bar and a "Create new Space" button. Below, a section titled "Spaces of the week" shows cards for various projects like "Parakeet Runnt 1.1b" (running on A106), "PASD Magnify" (running on ZERO), "IP-Adapter-FaceID" (running on A106), "M2UGen Demo" (running on A106), "AnyText" (running on A106), "Open-VL-Plus" (running on A106), "LongAnimateDiff" (running on A106), and "Open-Vocabulary SAM" (running on A106). Each card includes a thumbnail, the project name, the platform it's running on, the number of stars, and the last update time.

**Right Screenshot:** A screenshot of the GitHub "Trending" page. The header says "See what the GitHub community is most excited about today." It features tabs for "Repositories" and "Developers". The main area lists trending repositories such as "kaywrigley / chatbot-ui", "copilot-gpt4-service", "subql", and "meilisearch / meilisearch". Each repository card includes the owner, name, star count, language, and a brief description. The interface also includes filters for "Spoken Language", "Language", and "Date range", along with buttons for "Sponsor", "Star", and "Follow".



# Pytorch Lightning

- › Extends Pytorch to be faster, easier to use, and scale endlessly
- › Use to run experiments faster and easier

README    Code of conduct    Apache-2.0 license    Security



**Lightning**  
from the creators of Pytorch Lightning

The deep learning framework to pretrain, finetune and deploy AI models.

NEW- Lightning 2.0 features a clean and stable API!!

---

[Lightning.ai](#) • [PyTorch Lightning](#) • [Fabric](#) • [Lightning Apps](#) • [Docs](#) • [Community](#) • [Contribute](#) •

python 3.8 | 3.9 | 3.10 | 3.11 | pypi package 2.1.3 | downloads 4M/month | conda v2.1.3 | codecov 82%

chat 735 online | commit activity 19/week | License Apache 2.0

## Install Lightning

Simple installation from PyPI

```
pip install lightning
```



# Albumentations

- › Alter, enhance, or augment your images to your need
- › Understand your models and training better by using albumentations

## A Albumentations

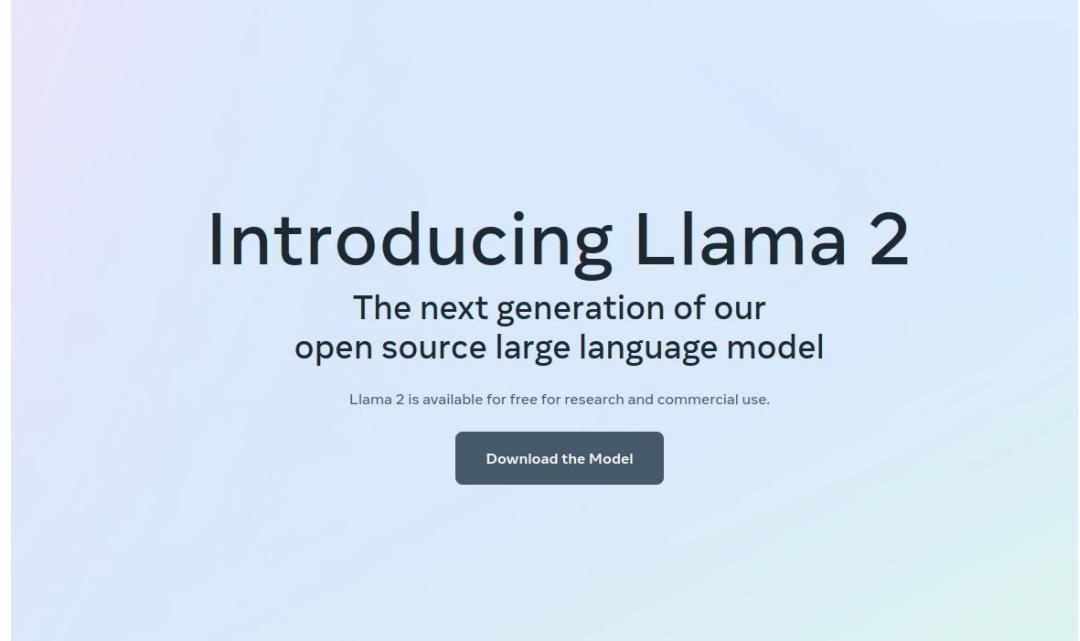
Fast and flexible image augmentations

<https://albumentations.ai>



# LLaMA2

- › Create from one of the worlds best LLMs available, all open source
- › Opportunities are endless, creativity is the main thing holding projects back



The image shows a landing page for Llama 2. The background is a gradient from light purple at the top to light blue at the bottom. In the center, the text "Introducing Llama 2" is displayed in a large, bold, black font. Below it, the text "The next generation of our open source large language model" is shown in a smaller, regular black font. At the bottom left, there is a small line of text: "Llama 2 is available for free for research and commercial use." To the right of this text is a dark grey rectangular button with the white text "Download the Model".

## Introducing Llama 2

The next generation of our  
open source large language model

Llama 2 is available for free for research and commercial use.

[Download the Model](#)



# Workshop Agenda

- > Overview
- > Installation & Environments
- > FiftyOne Basic Concepts
- > Open Source
- > **Next Steps**

# Next Steps > Resources

## Try FiftyOne

-  Google Colab, Jupyter, GitHub, or download a notebook

## Join us on FiftyOne community Slack

-  Join other data scientists solving interesting computer vision problems

## If you like our work, please star the FiftyOne project!

-  <https://github.com/voxel51/fiftyone>



# Next Steps > Delving Deeper

- › **Explore datasets** in the Dataset Zoo - video, geo, point clouds, ...
- › **Load a model** from the Model Zoo and use it to **add predictions** to a dataset
- › **Import** your own data into FiftyOne
- › **Compute and visualize embeddings** on your dataset
- › **Send a labeling job** to CVAT, Labelbox, or Label Studio via our integrations

