ultralytics / **ultralytics** Public

NEW - YOLOv8 🚀 in PyTorch > ONNX > OpenVINO > CoreML > TFLite

docs.ultralytics.com

AGPL-3.0 license

19.8k stars 4k forks Branches Tags Activity

Star Notifications

Code Issues 558 Pull requests 113 Discussions Actions

main Branches Tags Go to file Go to file Code

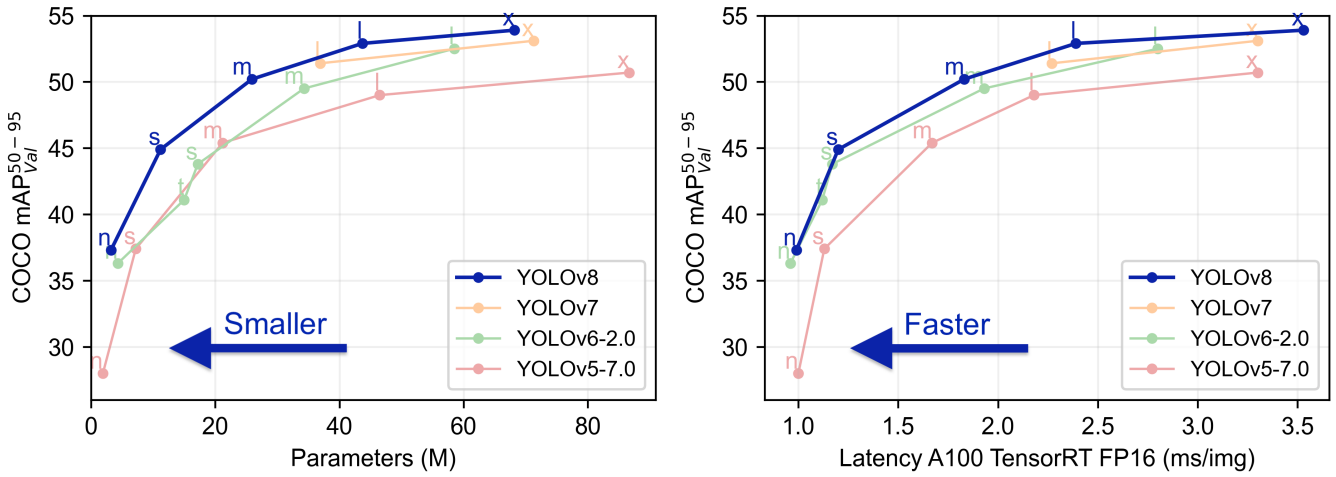
.github
docker
docs
examples
tests
ultralytics
.gitignore
.pre-commit-config.yaml
CITATION.cff
CONTRIBUTING.md
LICENSE
README.md
README.zh-CN.md
mkdocs.yml
pyproject.toml

README Code of conduct More

Ultralytics YOLOv8 is a cutting-edge, state-of-the-art (SOTA) model that builds upon the success of previous YOLO versions and introduces new features and improvements to further boost performance and flexibility. YOLOv8 is designed to be fast, accurate, and easy to use, making it an excellent choice for a wide range of object detection and tracking, instance segmentation, image classification and pose estimation tasks.

We hope that the resources here will help you get the most out of YOLOv8. Please browse the YOLOv8 Docs for details, raise an issue on GitHub for support, and join our Discord community for questions and discussions!

To request an Enterprise License please complete the form at Ultralytics Licensing.



Documentation

See below for a quickstart installation and usage example, and see the YOLOv8 Docs for full documentation on training, validation, prediction and deployment.

▼ Install

Pip install the ultralytics package including all requirements in a Python>=3.8 environment with PyTorch>=1.8.


```
pip install ultralytics
```

For alternative installation methods including Conda, Docker, and Git, please refer to the Quickstart Guide.

▼ Usage

CLI

YOLOv8 may be used directly in the Command Line Interface (CLI) with a `yolo` command:

```
yolo predict model=yolov8n.pt source='https://ultralytics.com/images/bus.jpg'
```

`yolo` can be used for a variety of tasks and modes and accepts additional arguments, i.e. `imgsz=640`. See the YOLOv8 CLI Docs for examples.

Python

YOLOv8 may also be used directly in a Python environment, and accepts the same arguments as in the CLI example above:

```python
from ultralytics import YOLO

# Load a model
model = YOLO("yolov8n.yaml")  # build a new model from scratch
model = YOLO("yolov8n.pt")  # load a pretrained model (recommended for training)

# Use the model
model.train(data="coco128.yaml", epochs=3)  # train the model
metrics = model.val()  # evaluate model performance on the validation set
results = model("https://ultralytics.com/images/bus.jpg")  # predict on an image
path = model.export(format="onnx")  # export the model to ONNX format
```

See YOLOv8 Python Docs for more examples.

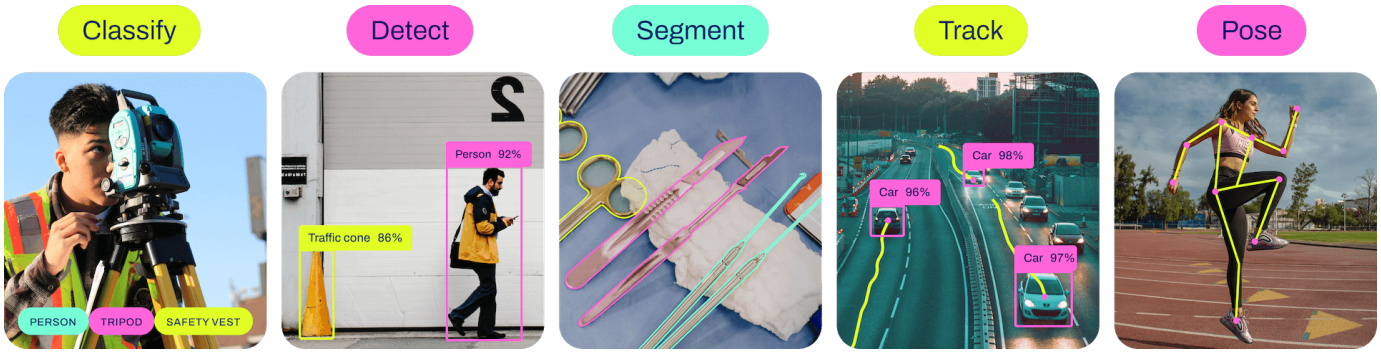
Notebooks

Ultralytics provides interactive notebooks for YOLOv8, covering training, validation, tracking, and more. Each notebook is paired with a YouTube tutorial, making it easy to learn and implement advanced YOLOv8 features.

Docs	Notebook	YouTube
YOLOv8 Train, Val, Predict and Export Modes	Open in Colab	
Ultralytics HUB QuickStart	Open in Colab	
YOLOv8 Multi-Object Tracking in Videos	Open in Colab	
YOLOv8 Object Counting in Videos	Open in Colab	
YOLOv8 Heatmaps in Videos	Open in Colab	
Ultralytics Datasets Explorer with SQL and OpenAI Integration New	Open in Colab	

Models

YOLOv8 Detect, Segment and Pose models pretrained on the COCO dataset are available here, as well as YOLOv8 Classify models pretrained on the ImageNet dataset. Track mode is available for all Detect, Segment and Pose models.



All Models download automatically from the latest Ultralytics release on first use.

▼ Detection (COCO)

See Detection Docs for usage examples with these models trained on COCO, which include 80 pre-trained classes.

Model	size (pixels)	mAP^{val} 50-95	Speed CPU ONNX (ms)	Speed A100 TensorRT (ms)	params (M)	FLOPs (B)
YOLOv8n	640	37.3	80.4	0.99	3.2	8.7
YOLOv8s	640	44.9	128.4	1.20	11.2	28.6
YOLOv8m	640	50.2	234.7	1.83	25.9	78.9

Model	size (pixels)	mAP^{val} 50-95	Speed CPU ONNX (ms)	Speed A100 TensorRT (ms)	params (M)	FLOPs (B)
YOLOv8l	640	52.9	375.2	2.39	43.7	165.2
YOLOv8x	640	53.9	479.1	3.53	68.2	257.8

- **mAP^{val}** values are for single-model single-scale on COCO val2017 dataset.
  Reproduce by `yolo val detect data=coco.yaml device=0`
- **Speed** averaged over COCO val images using an Amazon EC2 P4d instance.
  Reproduce by `yolo val detect data=coco.yaml batch=1 device=0|cpu`

▶ Detection (Open Image V7)

▶ Segmentation (COCO)

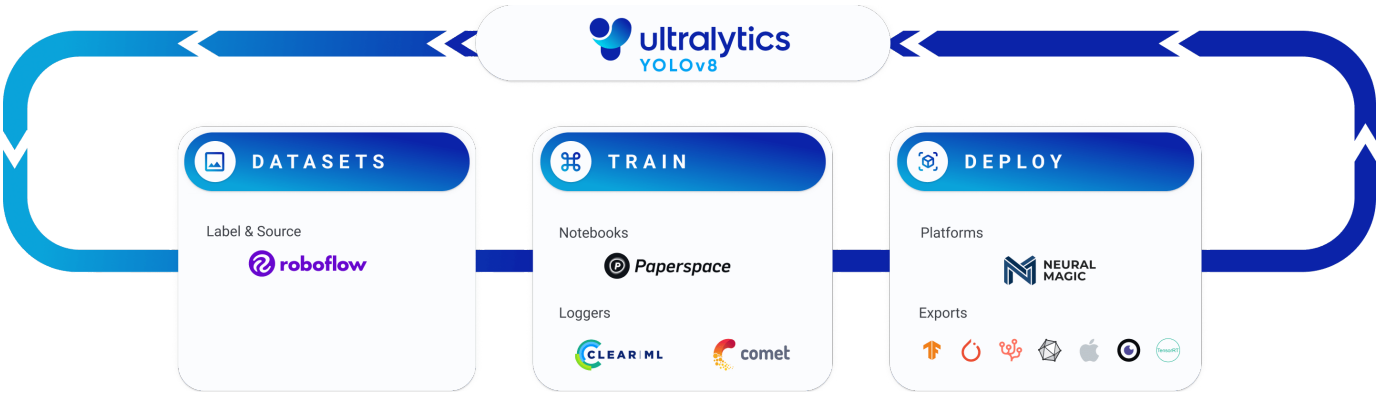
▶ Pose (COCO)

▶ OBB (DOTAv1)

▶ Classification (ImageNet)

Integrations

Our key integrations with leading AI platforms extend the functionality of Ultralytics' offerings, enhancing tasks like dataset labeling, training, visualization, and model management. Discover how Ultralytics, in collaboration with Roboflow, ClearML, Comet, Neural Magic and OpenVINO, can optimize your AI workflow.



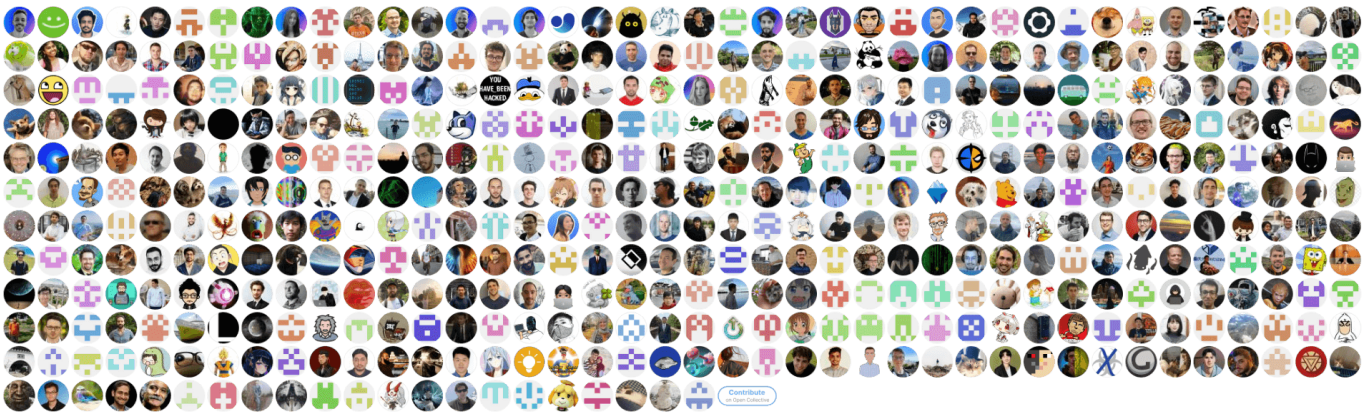
Roboflow	ClearML ⭐ NEW	Comet ⭐ NEW	Neural Magic ⭐ NEW
Label and export your custom datasets directly to YOLOv8 for training with Roboflow	Automatically track, visualize and even remotely train YOLOv8 using ClearML (open-source!)	Free forever, Comet lets you save YOLOv8 models, resume training, and interactively visualize and debug predictions	Run YOLOv8 inference up to 6x faster with Neural Magic DeepSparse

Ultralytics HUB

Experience seamless AI with Ultralytics HUB ⭐, the all-in-one solution for data visualization, YOLOv5 and YOLOv8 🚀 model training and deployment, without any coding. Transform images into actionable insights and bring your AI visions to life with ease using our cutting-edge platform and user-friendly Ultralytics App. Start your journey for **Free** now!

Contribute

We love your input! YOLOv5 and YOLOv8 would not be possible without help from our community. Please see our Contributing Guide to get started, and fill out our Survey to send us feedback on your experience. Thank you 🙏 to all our contributors!



License

Ultralytics offers two licensing options to accommodate diverse use cases:

- **AGPL-3.0 License**: This OSI-approved open-source license is ideal for students and enthusiasts, promoting open collaboration and knowledge sharing. See the LICENSE file for more details.
- **Enterprise License**: Designed for commercial use, this license permits seamless integration of Ultralytics software and AI models into commercial goods and services, bypassing the open-source requirements of AGPL-3.0. If your scenario involves embedding our solutions into a commercial offering, reach out through Ultralytics Licensing.

Releases 1

v8.1.0 - YOLOv8 Oriented Bounding Boxes (OBB) Latest
on Jan 10

Sponsor this project

patreon.com/ultralytics

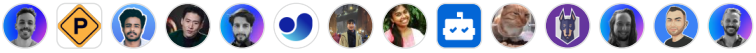
opencollective.com/ultralytics

Learn more about GitHub Sponsors

Used by 11.4k

 + 11,361

Contributors 252



+ 238 contributors

Languages

- **Python** 99.4%
- **Other** 0.6%