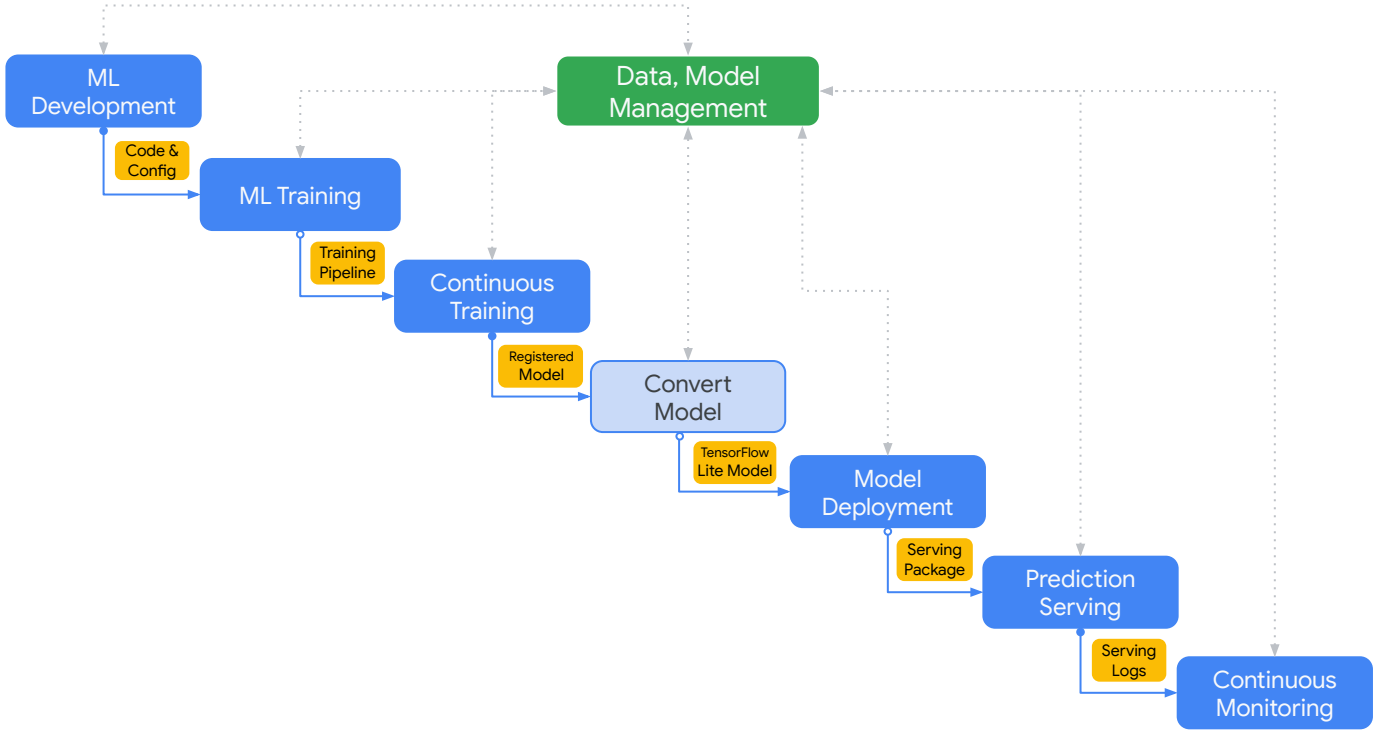
ML Frameworks and The Lay Of The Land

The MLOps Process for **TinyML**



The MLOps **Personas**

ML Engineer

ML Researcher

Data Scientist

Data Engineer

Software Engineer

DevOps

Business Analyst

ML Frameworks

TensorFlow

Google's **TensorFlow**

Open source ML framework

Released in 2015

PyTorch

Developed by **Facebook** AI Research

Open source ML framework

Released in 2017

Keras

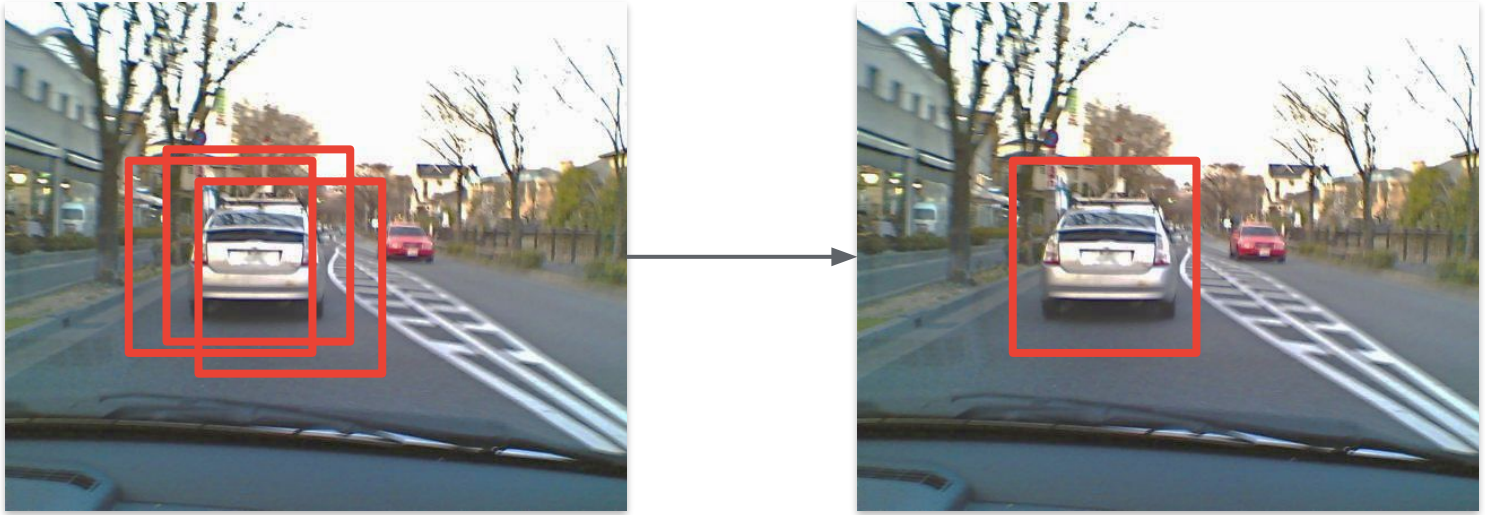
Primarily supported by **Google**

Open source DL framework

Released in 2015

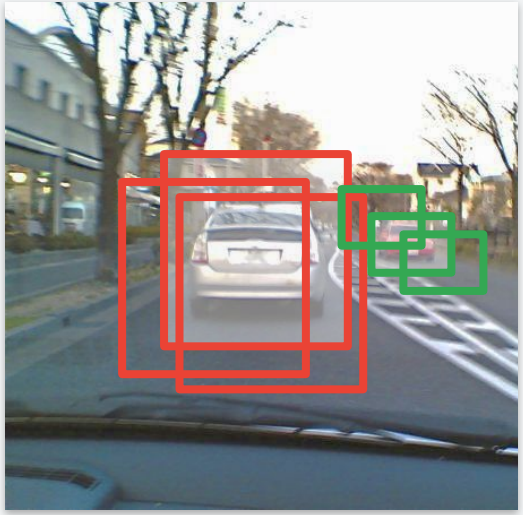
ML Framework	K		
Ease of use	**User-friendly**	Incomprehensive API	**Integrated with Python**
API Level	High-level API	Both High/Low-Level	**Low-level API**
Architecture	**Simple, readable, concise**	Not easy to use	Complex
Speed	Slow	**Fast**	**Fast**
Debugging	No need to debug	Difficult	**Helpful capabilities**
Creator	(Not sole library)	Google	Facebook

Non-Max Suppression



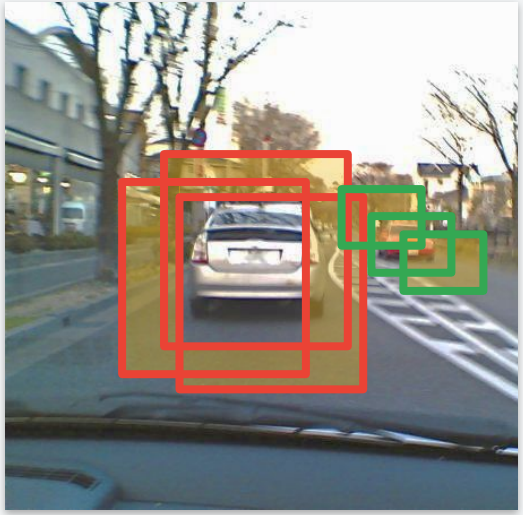
Non-Max Suppression

1. **Select boxes with highest objectiveness score**



Non-Max Suppression

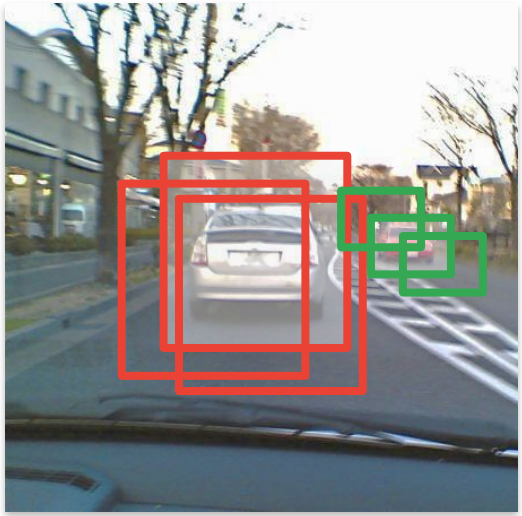
1. Select boxes with highest objectiveness score
2. **Compare overlap and remove boxes with >50% overlap**

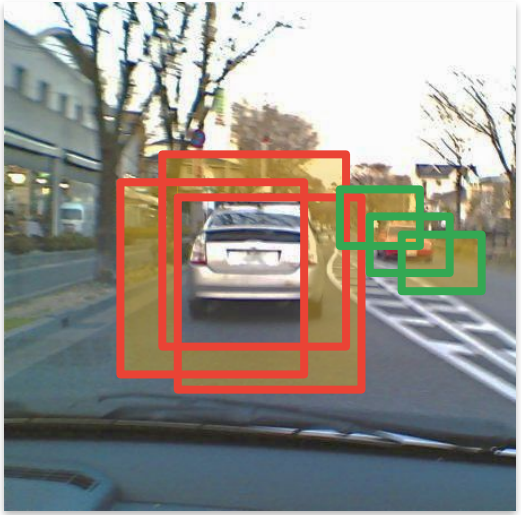


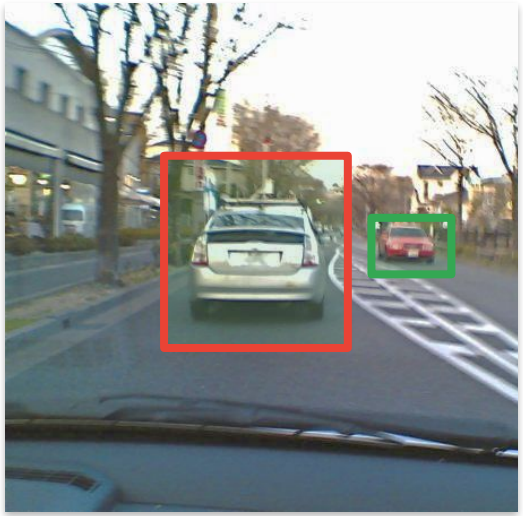
Non-Max Suppression

1. Select boxes with highest objectiveness score
2. Compare overlap and remove boxes with >50% overlap
3. **Move to next highest objectiveness score and return to Step 2.**

Non-Max Suppression







tf.image.non_max_suppression

TensorFlow 1 version | View source on GitHub

Greedily selects a subset of bounding boxes in descending order of score.

View aliases

```
tf.image.non_max_suppression(
    boxes, scores, max_output_size, iou_threshold=0.5,
    score_threshold=float('-inf'), name=None
)
```

Prunes away boxes that have high intersection-over-union (IOU) overlap with previously selected boxes. Bounding boxes are supplied as `[y1, x1, y2, x2]`, where `(y1, x1)` and `(y2, x2)` are the coordinates of any diagonal pair of box corners and the coordinates can be provided as normalized (i.e., lying in the interval `[0, 1]`) or absolute. Note that this algorithm is agnostic to where the origin is in the coordinate system. Note that this algorithm is invariant to orthogonal transformations and translations of the coordinate system; thus translating or reflections of the coordinate system result in the same boxes being selected by the algorithm. The output of this operation is a set of integers indexing into the input collection of bounding boxes representing the selected boxes. The bounding box coordinates corresponding to the selected indices can then be obtained using the `tf.gather` operation. For example:

```
selected_indices = tf.image.non_max_suppression(
    boxes, scores, max_output_size, iou_threshold)
selected_boxes = tf.gather(boxes, selected_indices)
```

`torchvision.ops.nms(boxes: torch.Tensor, scores: torch.Tensor, iou_threshold: float) → torch.Tensor` [SOURCE]

Performs non-maximum suppression (NMS) on the boxes according to their intersection-over-union (IoU).

NMS iteratively removes lower scoring boxes which have an IoU greater than iou_threshold with another (higher scoring) box.

If multiple boxes have the exact same score and satisfy the IoU criterion with respect to a reference box, the selected box is not guaranteed to be the same between CPU and GPU. This is similar to the behavior of argsort in PyTorch when repeated values are present.

Parameters

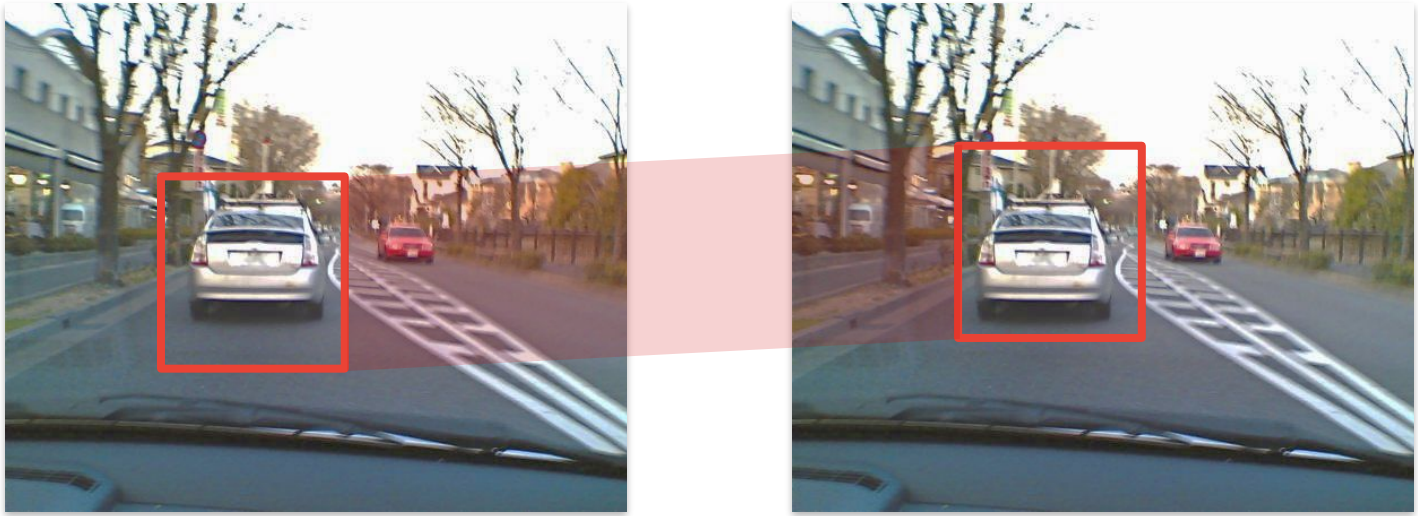
- **boxes** (*Tensor*[*N*, *4*]) – boxes to perform NMS on. They are expected to be in `(x1, y1, x2, y2)` format with `0 <= x1 < x2` and `0 <= y1 < y2`.
- **scores** (*Tensor*[*N*]) – scores for each one of the boxes
- **iou_threshold** (*float*) – discards all overlapping boxes with IoU > iou_threshold

Returns

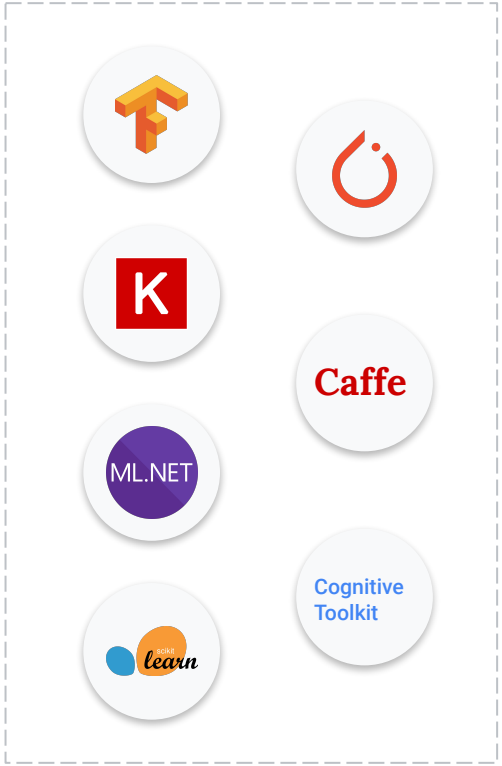
int64 tensor with the indices of the elements that have been kept by NMS, sorted in decreasing order of scores

Return type

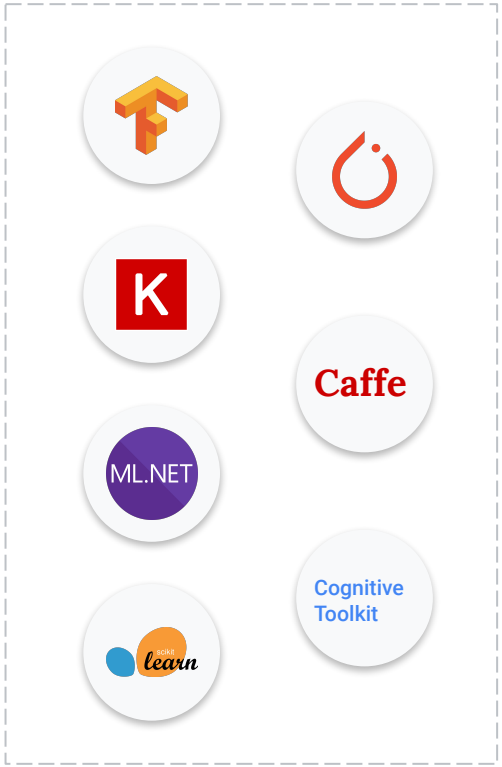
Tensor



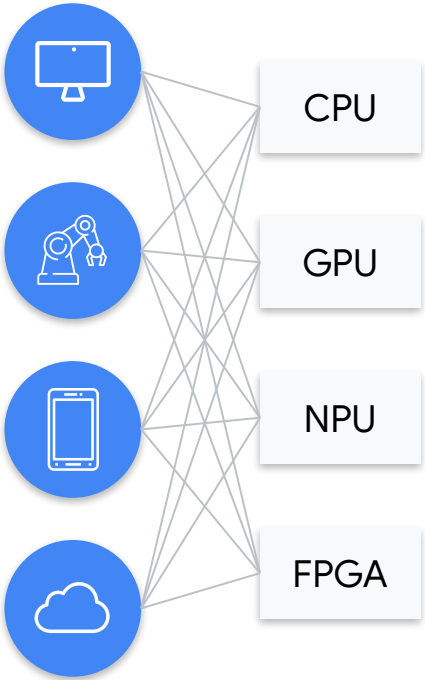
Training Framework

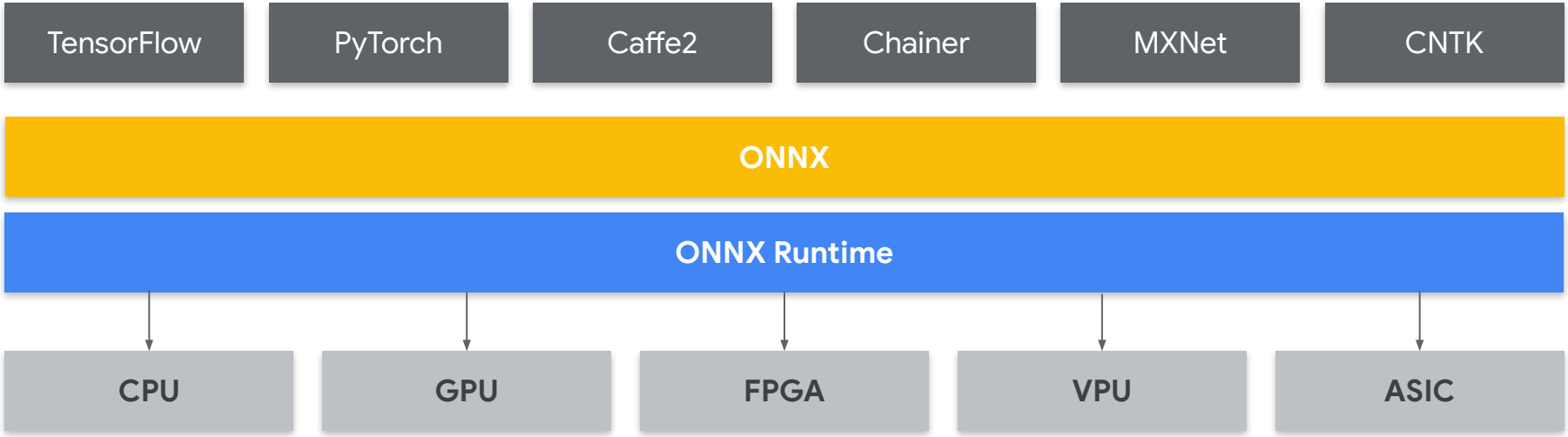


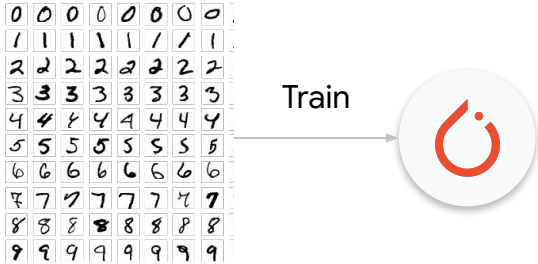
Training Framework

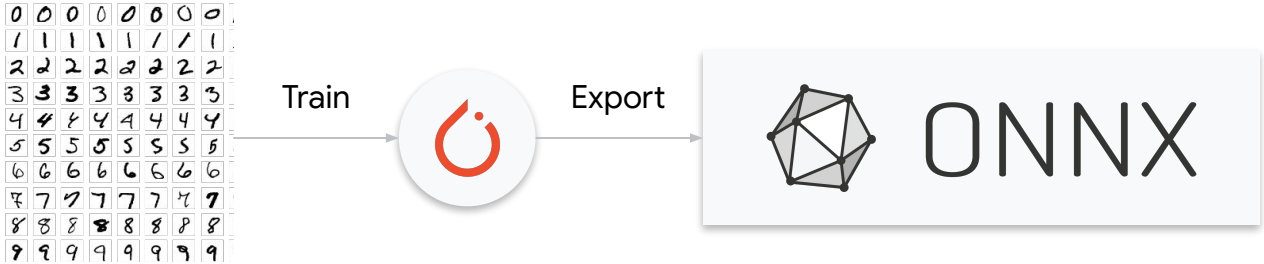


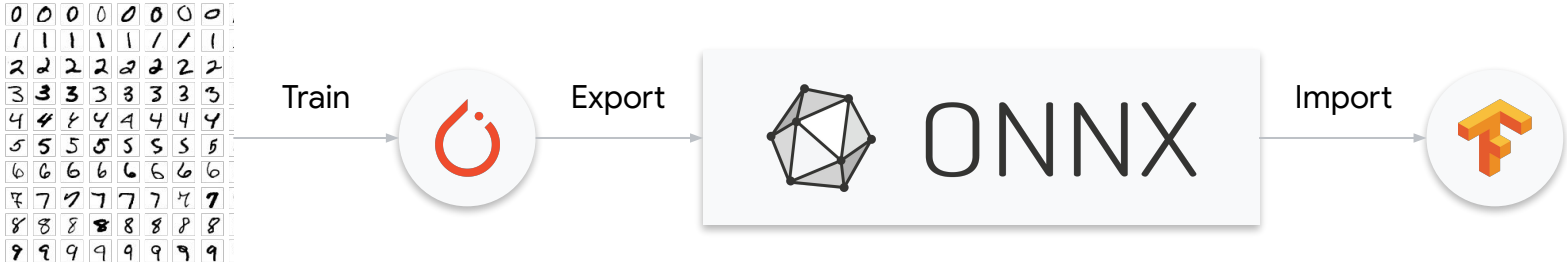
Deployment Target





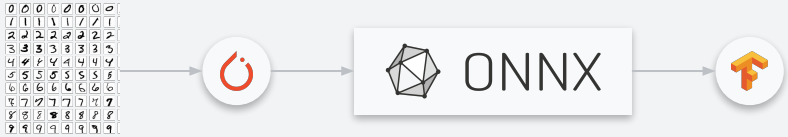






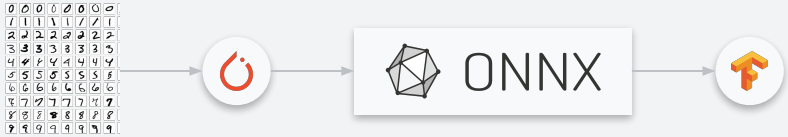
Export and Model Portability Issues

- Ops may (**not**) be supported evenly across all systems



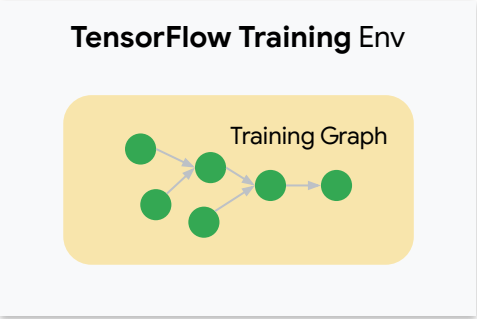
Export and Model Portability Issues

- Ops may (**not**) be supported evenly across all systems
- Frameworks implement their ***own*** library functions (e.g., NMS for detection)

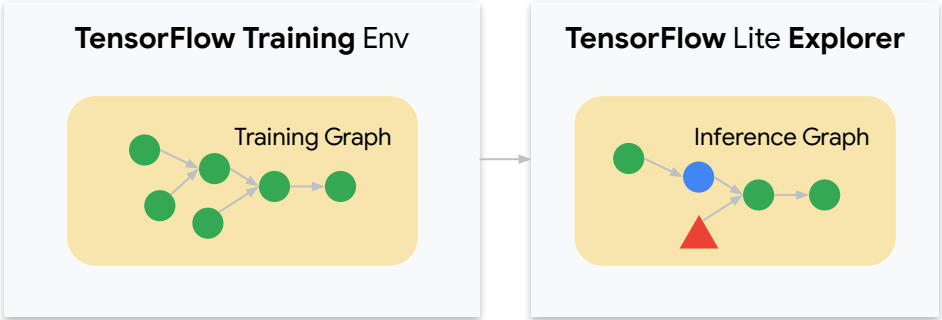


Benefits of Staying in the Same Ecosystem

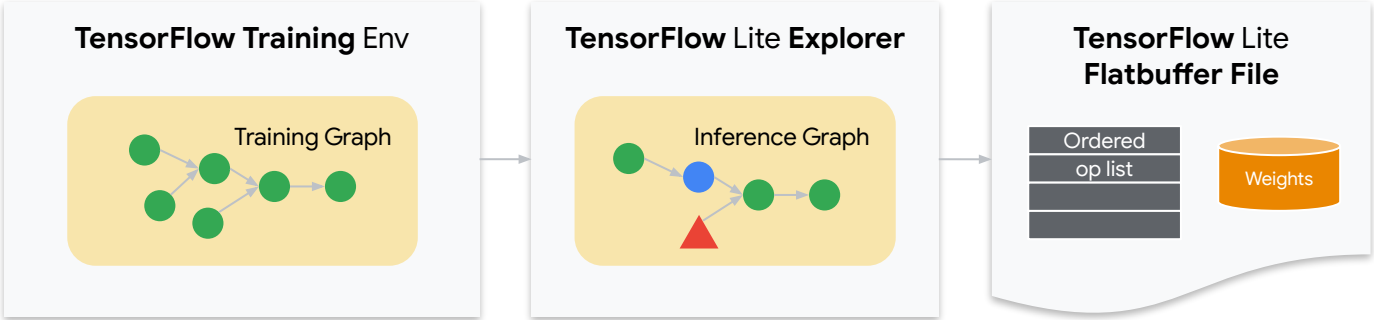
Benefits of Staying in the Same Ecosystem



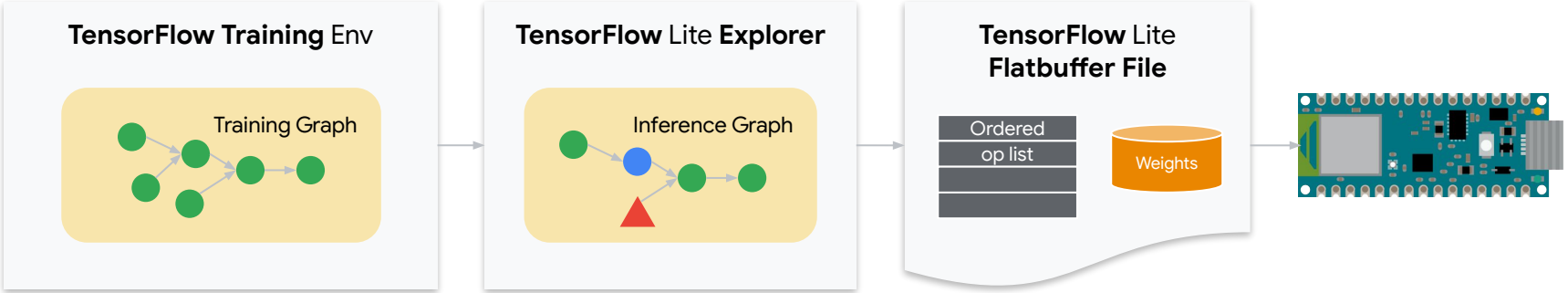
Benefits of Staying in the Same Ecosystem



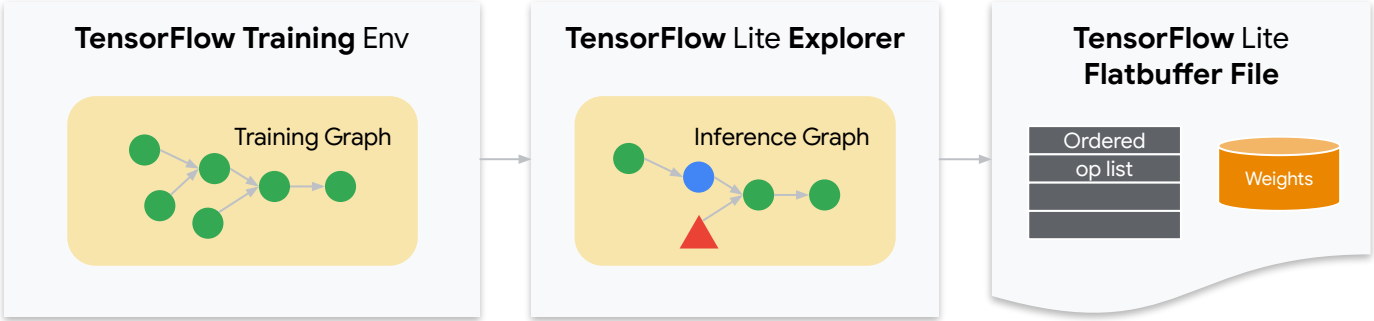
Benefits of Staying in the Same Ecosystem



Benefits of Staying in the Same Ecosystem



Benefits of Staying in the Same Ecosystem



- More **robust toolchain** for debugging and optimization
- Get to ***reuse*** the existing optimization infrastructure