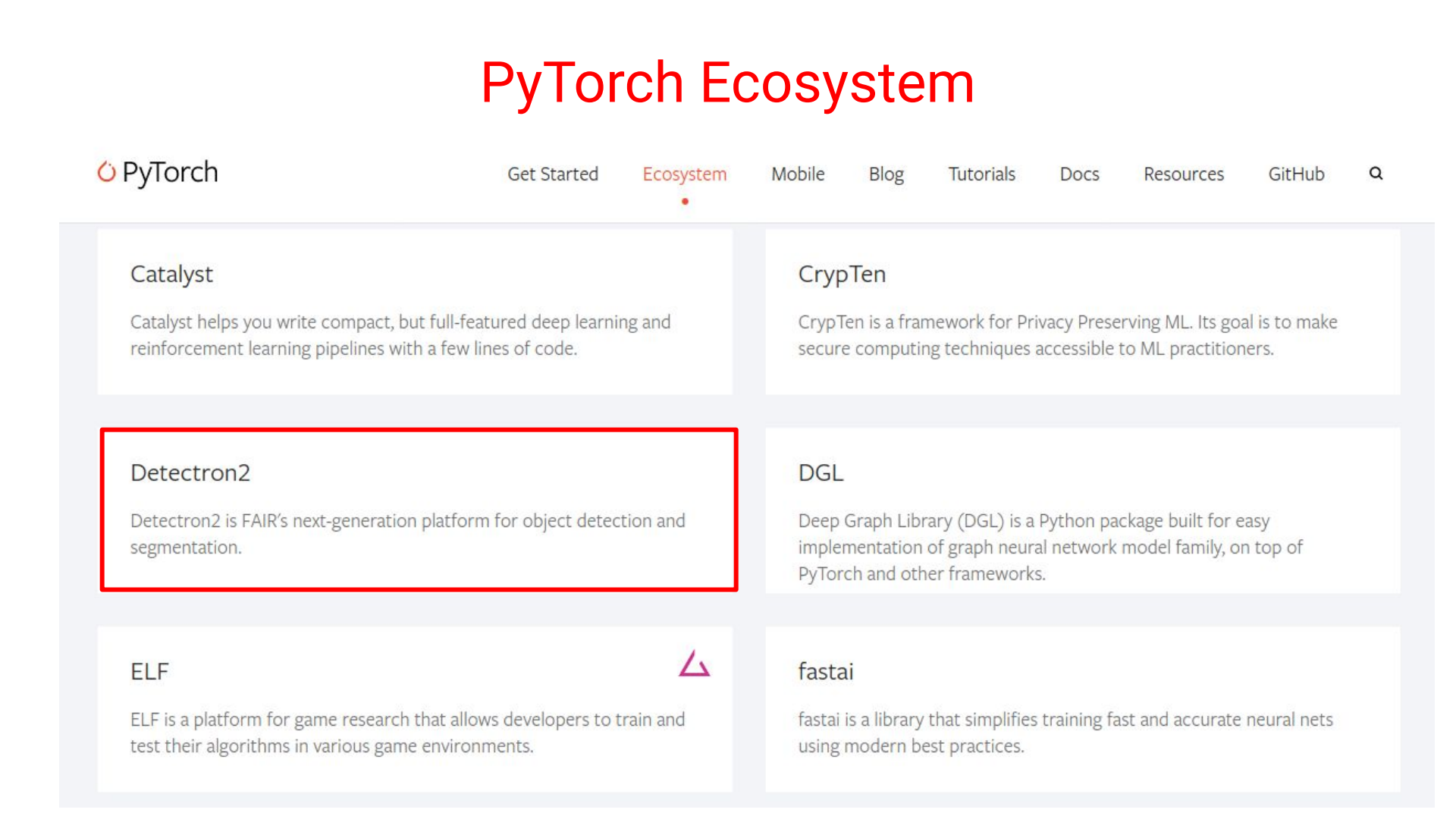
Implementing RetinaNet Using Detectron2

PyTorch Ecosystem



Detectron2

- Facebook AI Research's software system

Detectron2

- Facebook AI Research's software system
- Powered by the PyTorch deep learning framework

Detectron2

- Facebook AI Research's software system
- Powered by the PyTorch deep learning framework
- Implements state-of-the-art object detection and image segmentation algorithms

Important Modules in Detectron2

- data
- model_zoo
- config
- engine
- Utils.visualizer

Analytics Vidhya

Important Modules in Detectron2

- data
- model_zoo
- config
- engine
- Utils.visualizer

Analytics Vidhya

Important Modules in Detectron2 - data

- Use one of the existing datasets

Important Modules in Detectron2 - data

- Use one of the existing datasets
- To use custom dataset
  - Add dataset to existing ***DatasetCatalog***
  - Define related attributes (class names) ***MetadataCatalog***

Important Modules in Detectron2 - data

- Use one of the existing datasets
- To use custom dataset
  - Add dataset to existing ***DatasetCatalog***
  - Define related attributes (class names) ***MetadataCatalog***

```
## Adding custom dataset to the existing catalog
DatasetCatalog.register("bloodCellDet_dataset", bloodCellDet_Converter)


## Defining related attributes of the dataset
bloodCellDet_metadata = MetadataCatalog.get("bloodCellDet_dataset").set(thing_classes=["RBC", "WBC"])
```

Important Modules in Detectron2

- data
- model_zoo
- config
- engine
- Utils.visualizer

Important Modules in Detectron2 - model_zoo

- Consists of weights for pre-trained models

Important Modules in Detectron2 - model_zoo

- Consists of weights for pre-trained models
  - RPN
  - Fast R-CNN
  - Faster R-CNN
  - RetinaNet
  - Mask R-CNN

Analytics Vidhya

Important Modules in Detectron2 - model_zoo

- Consists of weights for pre-trained models
- Requires the following data format:

Important Modules in Detectron2 - model_zoo

- Consists of weights for pre-trained models
- Requires the following data format

  Dictionary to store
  - image ID
  - image width
  - image height
  - Target List

Analytics Vidhya

Important Modules in Detectron2 - model_zoo

- Consists of weights for pre-trained models
- Requires the following data format

  Dictionary to store
  - image ID
  - image width
  - image height
  - Target List
    - bounding box coordinates
    - respective class

Analytics Vidhya

Important Modules in Detectron2

- data
- model_zoo
- config
- engine
- Utils.visualizer

Analytics Vidhya

Important Modules in Detectron2 - config

- Consists default model configuration

```
_C.MODEL.LOAD_PROPOSALS = False
_C.MODEL.MASK_ON = False
_C.MODEL.KEYPOINT_ON = False
_C.MODEL.DEVICE = "cuda"
_C.MODEL.META_ARCHITECTURE = "GeneralizedRCNN"

# Path to a checkpoint file to be loaded to the model.
# You can find available models in the model zoo.
_C.MODEL.WEIGHTS = ""

# Default values are the mean pixel value from ImageNet
_C.MODEL.PIXEL_MEAN = [103.530, 116.280, 123.675]
```

Important Modules in Detectron2 - config

- Consists default model configuration
- Load default config file

```
_C.MODEL.LOAD_PROPOSALS = False
_C.MODEL.MASK_ON = False
_C.MODEL.KEYPOINT_ON = False
_C.MODEL.DEVICE = "cuda"
_C.MODEL.META_ARCHITECTURE = "GeneralizedRCNN"

# Path to a checkpoint file to be loaded to the model.
# You can find available models in the model zoo.
_C.MODEL.WEIGHTS = ""

# Default values are the mean pixel value from ImageNet
_C.MODEL.PIXEL_MEAN = [103.530, 116.280, 123.675]
```

Important Modules in Detectron2 - config

- Consists default model configuration
- Load default config file
- Update architecture as per requirement

```
_C.MODEL.LOAD_PROPOSALS = False
_C.MODEL.MASK_ON = False
_C.MODEL.KEYPOINT_ON = False
_C.MODEL.DEVICE = "cuda"
_C.MODEL.META_ARCHITECTURE = "RetinaNet"

# Path to a checkpoint file to be loaded to the model.
# You can find available models in the model zoo.
_C.MODEL.WEIGHTS = 'detectron2://ImageNetPretrained/MSRA/R-50.pkl'

# Default values are the mean pixel value from ImageNet
_C.MODEL.PIXEL_MEAN = [103.530, 116.280, 123.675]
```

Important Modules in Detectron2

- data
- model_zoo
- config
- engine
- Utils.visualizer

Analytics Vidhya

Important Modules in Detectron2 - engine

- For training and making predictions

```
trainer = DefaultTrainer(cfg)
```

Important Modules in Detectron2 - engine

- For training and making predictions

```
trainer = DefaultTrainer(cfg)

trainer.resume_or_load(resume = True)
```

- Load pretrain weight

Important Modules in Detectron2 - engine

- For training and making predictions
- Load pretrain weight
- Train the model for new data

```
trainer = DefaultTrainer(cfg)

trainer.resume_or_load(resume = True)

trainer.train()
```

Important Modules in Detectron2

- data
- model_zoo
- config
- engine
- Utils.visualizer

Analytics Vidhya

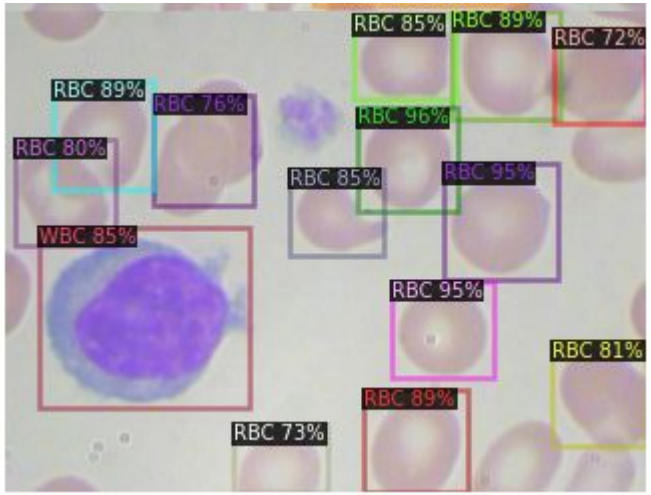
Important Modules in Detectron2 - visualizer

- Visualize data and bounding boxes

Analytics Vidhya

Important Modules in Detectron2 - visualizer

- Visualize data and bounding boxes
- Requires -
  - Input image
  - Box coordinates



Thank You

Analytics Vidhya

RetinaNet Details

- #### detectron
- #### used for
- #### data format required
- #### retinanet in detectron
- #### visualization

Important Modules in Detectron2 - visualizer

- Visualize data and bounding boxes
- Requires -
  - Input image
  - Box coordinates

Analytics Vidhya

Detectron2

- Consists of weights for pre-trained models

    a. RPN
    b. Fast R-CNN
    c. Faster R-CNN
    d. RetinaNet
    e. Mask R-CNN

using the following backbone network architectures:

- ResNeXt{50,101,152}
- ResNet{50,101,152}
- Feature Pyramid Networks (with ResNet/ResNeXt)
- VGG16