

Implementing RetinaNet Using Detectron2

PyTorch Ecosystem

[Get Started](#)[Ecosystem](#)[Mobile](#)[Blog](#)[Tutorials](#)[Docs](#)[Resources](#)[GitHub](#)

Catalyst

Catalyst helps you write compact, but full-featured deep learning and reinforcement learning pipelines with a few lines of code.

CrypTen

CrypTen is a framework for Privacy Preserving ML. Its goal is to make secure computing techniques accessible to ML practitioners.

Detectron2

Detectron2 is FAIR's next-generation platform for object detection and segmentation.

DGL

Deep Graph Library (DGL) is a Python package built for easy implementation of graph neural network model family, on top of PyTorch and other frameworks.

ELF

ELF is a platform for game research that allows developers to train and test their algorithms in various game environments.



fastai

fastai is a library that simplifies training fast and accurate neural nets using modern best practices.

Detectron2

- Facebook AI Research's software system



Detectron2



Detectron2

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



Detectron2

Analytics
Vidhya

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



Important Modules in Detectron2

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



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](#)

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

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

Important Modules in Detectron2

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



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

```
_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]
```

- Update architecture as per requirement

Important Modules in Detectron2

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



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)
```


- Load pretrain weight

```
trainer.resume_or_load(resume = True)
```

Important Modules in Detectron2 - engine

- For training and making predictions

- Load pretrain weight



```
trainer = DefaultTrainer(cfg)
trainer.resume_or_load(resume = True)
trainer.train()
```

- Train the model for new data

Important Modules in Detectron2

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



Important Modules in Detectron2 - visualizer

- Visualize data and bounding boxes

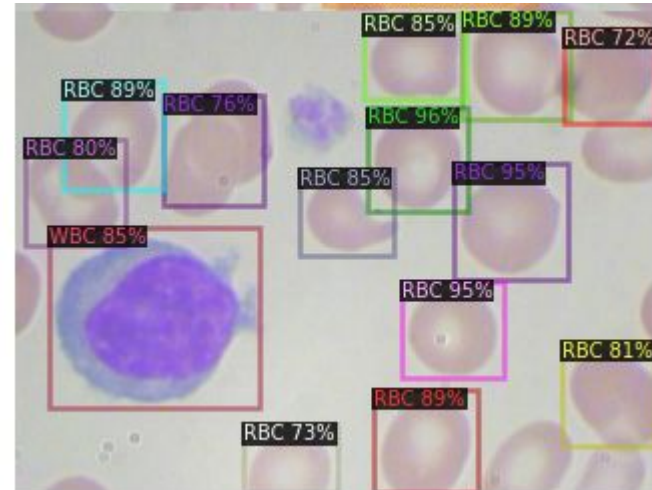


Important Modules in Detectron2 - visualizer

- Visualize data and bounding boxes

- Requires -

- Input image
- Box coordinates





Thank You

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



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](#)