

Deep Learning Homework 3

Siddharth Choudhary sc7530

1. *Convolutional layers*

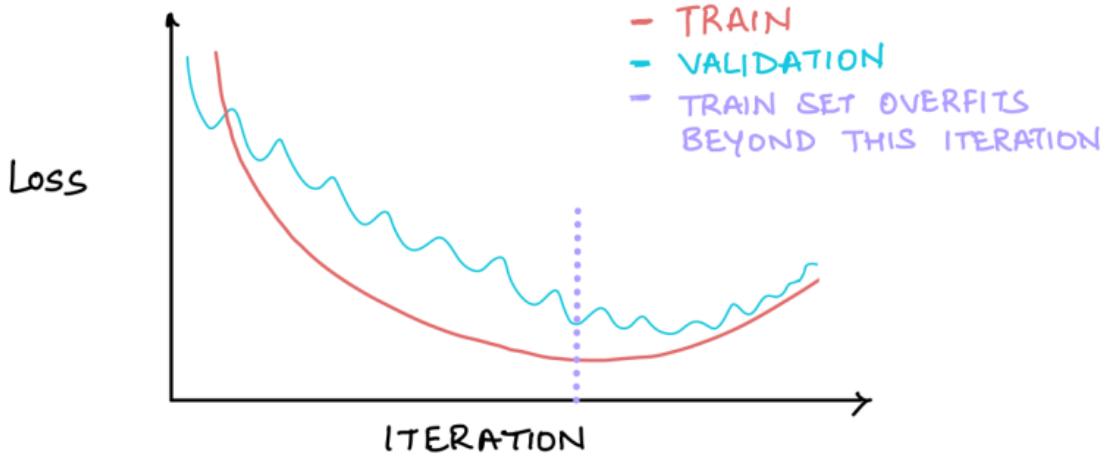
- a. Shapes of both Z and Y are same which is :

$$Z[i, j, m] = Y[i, j, m] = [46 \times 62 \times 20]$$

- b. The number of input channels are 10 and,
The number of output channels are 20
- c. The total number of trainable parameters are given by the sum of total number of weights and the biases which is given by : $(3 \times 3 \times 10 \times 20) + 20 = 1820$ trainable parameters.
- d. The total number of arithmetic multiplications needed to perform are $46 \times 62 \times 10 \times 20 \times 3 \times 3 = 5133600$

2. Overfitting in neural nets.

- a. A sketch of typical curves of loss versus iteration count is given below :

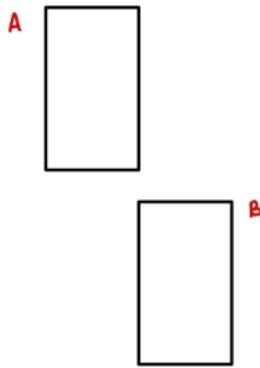


- b. Apart from data augmentation, Regularization is another technique to combat overfitting which reduces the complexity of the model by adding a penalty term to the loss function and constraining the weights. Early stopping can also be used which helps in determining how many iterations to run before the model begins to over-fit.
- c. In case of a cat-vs-dog classifier, using data augmentation techniques like scaling, rotating and so on works well because of the fact that intuitively, no matter how much a cat or a dog is rotated or scaled they still remain a cat or dog respectively. Thus increasing the size of data-set using data augmentation can be used for an image classification task to reduce over-fitting on the training set. But in an image classification task for identifying English handwritten characters data augmentation will not work since there are several alphabets that can be obtained by simply shearing, rotating or mirroring another alphabet. Moreover, with billions of people in the world, each with their own unique handwriting style and pattern, some alphabets can easily be wrongly predicted, thus data augmentation can actually make things worse in such a classification task.

3. The IoU metric

- For two bounding boxes present in the same plane, there are three possible cases that exist.

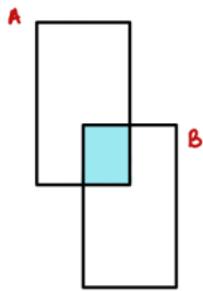
Case 1 :



When both the bounding boxes do not have any overlapping area. In such a situation, the Intersection over Union is given by:

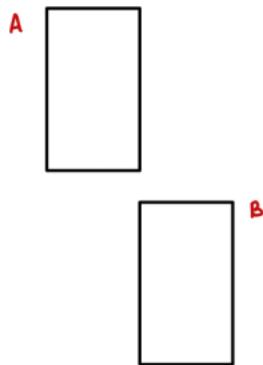
$$IOU = \frac{A \cap B}{A \cup B} = 0$$

Case 2 :



When both the bounding boxes have some overlapping area $|A \cap B| < |A \cup B|$

Case 3 :



When the bounding boxes completely overlap each other. In such a situation, the Intersection over Union is given by:

$$IOU = \frac{A \cap B}{A \cup B} = \frac{A \cup B}{A \cup B} = 1$$

From the above three cases present for the bounding boxes we can imply that the Intersection over Union is always a non negative real number in the range [0, 1] as shown in Case 1 and 2.

- b. As shown in the above part, the Intersection over Union metric is always a non-negative real constant. Since the gradient descent optimization technique requires the function to be differentiable, and a constant cannot be differentiated, we can say that the IoU metric cannot be directly optimized with gradient descent. Another way to argue this would be to show in the above part in Case 1. When there is no intersection between the boxes, the IoU metric is always 0. Since gradient at 0 cannot give us a direction to move and the case 1 possibility occurs often, gradient descent cannot be directly used for optimizing the IoU metric.

```
[ ]: !pip install pyyaml==5.1 pycocotools>=2.0.1
import torch, torchvision
print(torch.__version__, torch.cuda.is_available())
!gcc --version
assert torch.__version__.startswith("1.6")
!pip install detectron2 -f https://dl.fbaipublicfiles.com/detectron2/wheels/
→cu101/torch1.6/index.html
```

```
1.6.0+cu101 True
gcc (Ubuntu 7.5.0-3ubuntu1~18.04) 7.5.0
Copyright (C) 2017 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```

```
Looking in links:
https://dl.fbaipublicfiles.com/detectron2/wheels/cu101/torch1.6/index.html
Collecting detectron2
  Downloading https://dl.fbaipublicfiles.com/detectron2/wheels/cu101/torch
1.6/detectron2-0.2.1%2Bcu101-cp36-cp36m-linux_x86_64.whl (6.6MB)
    || 6.6MB 398kB/s
Collecting mock
  Downloading https://files.pythonhosted.org/packages/cd/74/d72daf8dff5b6566db85
7cf088907bb0355f5dd2914c4b3ef065c790735/mock-4.0.2-py3-none-any.whl
Requirement already satisfied: tqdm>4.29.0 in /usr/local/lib/python3.6/dist-
packages (from detectron2) (4.41.1)
Collecting yacs>=0.1.6
  Downloading https://files.pythonhosted.org/packages/38/4f/fe9a4d472aa867878ce3
bb7efb16654c5d63672b86dc0e6e953a67018433/yacs-0.1.8-py3-none-any.whl
Collecting fvcore>=0.1.1
  Downloading https://files.pythonhosted.org/packages/8f/14/3d359bd5526262b15dfb
b471cc1680a6aa384ed5883f0455c859f9b4224e/fvcore-0.1.2.post20201016.tar.gz
Requirement already satisfied:云cloudpickle in /usr/local/lib/python3.6/dist-
packages (from detectron2) (1.3.0)
Requirement already satisfied: pycocotools>=2.0.1 in
/usr/local/lib/python3.6/dist-packages (from detectron2) (2.0.2)
Requirement already satisfied: tensorboard in /usr/local/lib/python3.6/dist-
packages (from detectron2) (2.3.0)
Requirement already satisfied: tabulate in /usr/local/lib/python3.6/dist-
packages (from detectron2) (0.8.7)
Requirement already satisfied: termcolor>=1.1 in /usr/local/lib/python3.6/dist-
packages (from detectron2) (1.1.0)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.6/dist-
packages (from detectron2) (3.2.2)
Collecting Pillow>=7.1
  Downloading https://files.pythonhosted.org/packages/a6/9c/829c74f7f7f129
616dfb6f75af72529f895d29db71c7aeb46b02fcfb26d/Pillow-8.0.0-cp36-cp36m-manylinux
1_x86_64.whl (2.2MB)
```

```
|| 2.2MB 17.3MB/s
Requirement already satisfied: pydot in /usr/local/lib/python3.6/dist-
packages (from detectron2) (1.3.0)
Requirement already satisfied: future in /usr/local/lib/python3.6/dist-packages
(from detectron2) (0.16.0)
Requirement already satisfied: PyYAML in /usr/local/lib/python3.6/dist-packages
(from yacs>=0.1.6->detectron2) (5.1)
Requirement already satisfied: numpy in /usr/local/lib/python3.6/dist-packages
(from fvcore>=0.1.1->detectron2) (1.18.5)
Collecting portalocker
  Downloading https://files.pythonhosted.org/packages/89/a6/3814b7107e0788040870
e8825eebf214d72166adf656ba7d4bf14759a06a/portalocker-2.0.0-py2.py3-none-any.whl
Requirement already satisfied: cython>=0.27.3 in /usr/local/lib/python3.6/dist-
packages (from pycocotools>=2.0.1->detectron2) (0.29.21)
Requirement already satisfied: setuptools>=18.0 in
/usr/local/lib/python3.6/dist-packages (from pycocotools>=2.0.1->detectron2)
(50.3.0)
Requirement already satisfied: grpcio>=1.24.3 in /usr/local/lib/python3.6/dist-
packages (from tensorboard->detectron2) (1.32.0)
Requirement already satisfied: markdown>=2.6.8 in /usr/local/lib/python3.6/dist-
packages (from tensorboard->detectron2) (3.2.2)
Requirement already satisfied: werkzeug>=0.11.15 in
/usr/local/lib/python3.6/dist-packages (from tensorboard->detectron2) (1.0.1)
Requirement already satisfied: protobuf>=3.6.0 in /usr/local/lib/python3.6/dist-
packages (from tensorboard->detectron2) (3.12.4)
Requirement already satisfied: google-auth<2,>=1.6.3 in
/usr/local/lib/python3.6/dist-packages (from tensorboard->detectron2) (1.17.2)
Requirement already satisfied: six>=1.10.0 in /usr/local/lib/python3.6/dist-
packages (from tensorboard->detectron2) (1.15.0)
Requirement already satisfied: requests<3,>=2.21.0 in
/usr/local/lib/python3.6/dist-packages (from tensorboard->detectron2) (2.23.0)
Requirement already satisfied: tensorboard-plugin-wit>=1.6.0 in
/usr/local/lib/python3.6/dist-packages (from tensorboard->detectron2) (1.7.0)
Requirement already satisfied: google-auth-oauthlib<0.5,>=0.4.1 in
/usr/local/lib/python3.6/dist-packages (from tensorboard->detectron2) (0.4.1)
Requirement already satisfied: wheel>=0.26; python_version >= "3" in
/usr/local/lib/python3.6/dist-packages (from tensorboard->detectron2) (0.35.1)
Requirement already satisfied: absl-py>=0.4 in /usr/local/lib/python3.6/dist-
packages (from tensorboard->detectron2) (0.10.0)
Requirement already satisfied: kiwisolver>=1.0.1 in
/usr/local/lib/python3.6/dist-packages (from matplotlib->detectron2) (1.2.0)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.6/dist-
packages (from matplotlib->detectron2) (0.10.0)
Requirement already satisfied: pyparsing!=2.0.4,!>=2.1.2,!>=2.1.6,>=2.0.1 in
/usr/local/lib/python3.6/dist-packages (from matplotlib->detectron2) (2.4.7)
Requirement already satisfied: python-dateutil>=2.1 in
/usr/local/lib/python3.6/dist-packages (from matplotlib->detectron2) (2.8.1)
Requirement already satisfied: importlib-metadata; python_version < "3.8" in
```

```
/usr/local/lib/python3.6/dist-packages (from
markdown>=2.6.8->tensorboard->detectron2) (2.0.0)
Requirement already satisfied: pyasn1-modules>=0.2.1 in
/usr/local/lib/python3.6/dist-packages (from google-
auth<2,>=1.6.3->tensorboard->detectron2) (0.2.8)
Requirement already satisfied: cachetools<5.0,>=2.0.0 in
/usr/local/lib/python3.6/dist-packages (from google-
auth<2,>=1.6.3->tensorboard->detectron2) (4.1.1)
Requirement already satisfied: rsa<5,>=3.1.4; python_version >= "3" in
/usr/local/lib/python3.6/dist-packages (from google-
auth<2,>=1.6.3->tensorboard->detectron2) (4.6)
Requirement already satisfied: chardet<4,>=3.0.2 in
/usr/local/lib/python3.6/dist-packages (from
requests<3,>=2.21.0->tensorboard->detectron2) (3.0.4)
Requirement already satisfied: urllib3!=1.25.0,!>=1.25.1,<1.26,>=1.21.1 in
/usr/local/lib/python3.6/dist-packages (from
requests<3,>=2.21.0->tensorboard->detectron2) (1.24.3)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.6/dist-packages (from
requests<3,>=2.21.0->tensorboard->detectron2) (2020.6.20)
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.6/dist-
packages (from requests<3,>=2.21.0->tensorboard->detectron2) (2.10)
Requirement already satisfied: requests-oauthlib>=0.7.0 in
/usr/local/lib/python3.6/dist-packages (from google-auth-
oauthlib<0.5,>=0.4.1->tensorboard->detectron2) (1.3.0)
Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.6/dist-
packages (from importlib-metadata; python_version <
"3.8"->markdown>=2.6.8->tensorboard->detectron2) (3.2.0)
Requirement already satisfied: pyasn1<0.5.0,>=0.4.6 in
/usr/local/lib/python3.6/dist-packages (from pyasn1-modules>=0.2.1->google-
auth<2,>=1.6.3->tensorboard->detectron2) (0.4.8)
Requirement already satisfied: oauthlib>=3.0.0 in /usr/local/lib/python3.6/dist-
packages (from requests-oauthlib>=0.7.0->google-auth-
oauthlib<0.5,>=0.4.1->tensorboard->detectron2) (3.1.0)
Building wheels for collected packages: fvcore
  Building wheel for fvcore (setup.py) ... done
  Created wheel for fvcore: filename=fvcore-0.1.2.post20201016-cp36-none-any.whl
size=44196
sha256=f14e37f698046a6635e79f677e690ffa506c2ffea6d3d7b6106b20c4c592a7ee
  Stored in directory: /root/.cache/pip/wheels/f3/3f/35/86873c1dde45a9fb1ba7921
232ea15c570165a9d4f4d831c7
Successfully built fvcore
ERROR: albumentations 0.1.12 has requirement imgaug<0.2.7,>=0.2.5, but
you'll have imgaug 0.2.9 which is incompatible.
Installing collected packages: mock, yacs, portalocker, Pillow, fvcore,
detectron2
  Found existing installation: Pillow 7.0.0
```

```
Uninstalling Pillow-7.0.0:  
  Successfully uninstalled Pillow-7.0.0  
Successfully installed Pillow-8.0.0 detectron2-0.2.1+cu101  
fvcore-0.1.2.post20201016 mock-4.0.2 portalocker-2.0.0 yacs-0.1.8
```

```
[ ]: from detectron2 import model_zoo  
from detectron2.engine import DefaultPredictor  
from detectron2.config import get_cfg  
from detectron2.utils.visualizer import Visualizer  
from detectron2.data import MetadataCatalog  
import cv2  
from google.colab.patches import cv2_imshow  
  
im = cv2.imread('./central-park-balloon-man-madeline-ellis.jpg')  
cv2_imshow(im)
```



```
[ ]: cfg = get_cfg()  
cfg.merge_from_file(model_zoo.get_config_file("COCO-Keypoints/  
    ↳keypoint_rcnn_R_50_FPN_1x.yaml"))  
cfg.MODEL.ROI_HEADS.SCORE_THRESH_TEST = 0.45  
cfg.MODEL.WEIGHTS = "detectron2://COCO-Keypoints/keypoint_rcnn_R_50_FPN_1x/  
    ↳137261548/model_final_04e291.pkl"  
predictor = DefaultPredictor(cfg)
```

```

outputs = predictor(im)

v = Visualizer(im[:, :, ::-1], MetadataCatalog.get(cfg.DATASETS.TRAIN[0]), scale=1.2)
v = v.draw_instance_predictions(outputs["instances"].to("cpu"))
cv2_imshow(v.get_image()[:, :, ::-1])

```



```

[ ]: cfg = get_cfg()
cfg.merge_from_file(model_zoo.get_config_file("COCO-InstanceSegmentation/
→mask_rcnn_R_50_FPN_3x.yaml"))
cfg.MODEL.ROI_HEADS.SCORE_THRESH_TEST = 0.49
cfg.MODEL.WEIGHTS = "detectron2://COCO-InstanceSegmentation/
→mask_rcnn_R_50_FPN_3x/137849600/model_final_f10217.pkl"

predictor = DefaultPredictor(cfg)
outputs = predictor(im)

v = Visualizer(im[:, :, ::-1], MetadataCatalog.get(cfg.DATASETS.TRAIN[0]), scale=1.2)
v = v.draw_instance_predictions(outputs["instances"].to("cpu"))
cv2_imshow(v.get_image()[:, :, ::-1])

```



```
[ ]: !wget https://github.com/matterport/Mask_RCNN/releases/download/v2.1/
      ↳balloon_dataset.zip
!unzip balloon_dataset.zip > /dev/null
```

```
--2020-10-16 18:16:15-- https://github.com/matterport/Mask_RCNN/releases/download/v2.1/balloon_dataset.zip
Resolving github.com (github.com)... 13.229.188.59
Connecting to github.com (github.com)|13.229.188.59|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://github-production-release-asset-2e65be.s3.amazonaws.com/107595
270/737339e2-2b83-11e8-856a-188034eb3468?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-
Credential=AKIAIWNJYAX4CSVEH53A%2F20201016%2Fus-
east-1%2Fs3%2Faws4_request&X-Amz-Date=20201016T181615Z&X-Amz-Expires=300&X-Amz-S
ignature=89f8474db2ec8d026206ac264ce2e2b2d373e8d4072a654107e74518b72aba1f&X-Amz-
SignedHeaders=host&actor_id=0&key_id=0&repo_id=107595270&response-content-
disposition=attachment%3B%20filename%3Dballoon_dataset.zip&response-content-
type=application%2Foctet-stream [following]
--2020-10-16 18:16:15-- https://github-production-release-asset-2e65be.s3.amazo
naws.com/107595270/737339e2-2b83-11e8-856a-188034eb3468?X-Amz-
Algorithm=AWS4-HMAC-SHA256&X-Amz-
Credential=AKIAIWNJYAX4CSVEH53A%2F20201016%2Fus-
east-1%2Fs3%2Faws4_request&X-Amz-Date=20201016T181615Z&X-Amz-Expires=300&X-Amz-S
ignature=89f8474db2ec8d026206ac264ce2e2b2d373e8d4072a654107e74518b72aba1f&X-Amz-
```

```

SignedHeaders=host&actor_id=0&key_id=0&repo_id=107595270&response-content-
disposition=attachment%3B%20filename%3Dballoon_dataset.zip&response-content-
type=application%2Foctet-stream
Resolving github-production-release-asset-2e65be.s3.amazonaws.com (github-
production-release-asset-2e65be.s3.amazonaws.com) ... 52.217.12.124
Connecting to github-production-release-asset-2e65be.s3.amazonaws.com (github-
production-release-asset-2e65be.s3.amazonaws.com)|52.217.12.124|:443...
connected.
HTTP request sent, awaiting response... 200 OK
Length: 38741381 (37M) [application/octet-stream]
Saving to: 'balloon_dataset.zip'

balloon_dataset.zip 100%[=====] 36.95M 9.12MB/s in 4.1s

2020-10-16 18:16:20 (9.12 MB/s) - 'balloon_dataset.zip' saved
[38741381/38741381]

```

```

[ ]: import os
import numpy as np
import json
from detectron2.structures import BoxMode
from detectron2.data import DatasetCatalog, MetadataCatalog
import itertools

def get_balloon_dicts(img_dir):

    json_file = os.path.join(img_dir, 'via_region_data.json')

    with open(json_file) as f:
        imgs_anns = json.load(f)

    dataset_dicts = []

    for _, v in imgs_anns.items():

        record = {}

        filename = os.path.join(img_dir, v['filename'])
        height, width = cv2.imread(filename).shape[:2]

        record['file_name'] = filename
        record['height'] = height
        record['width'] = width

        annos = v['regions']
        objs = []

```

```

        for _, anno in annos.items():

            assert not anno['region_attributes']
            anno = anno['shape_attributes']
            px = anno['all_points_x']
            py = anno['all_points_y']
            poly = [(x + 0.5, y + 0.5) for x, y in zip(px, py)]
            poly = list(itertools.chain.from_iterable(poly))

            obj = {'bbox' : [np.min(px), np.min(py), np.max(px), np.max(py)],
                   'bbox_mode' : BoxMode.XYXY_ABS,
                   'segmentation' : [poly],
                   'category_id' : 0,
                   'iscrowd' : 0}
            objs.append(obj)

        record['annotations'] = objs
        dataset_dicts.append(record)

    return dataset_dicts

for d in ['train', 'val']:

    DatasetCatalog.register('balloon/' + d, lambda d=d:_get_balloon_dicts('balloon/' + d))
    MetadataCatalog.get('balloon/' + d).set(thing_classes = ['balloon'])

balloon_metadata = MetadataCatalog.get('balloon/train')

```

```

[ ]: from detectron2.engine import DefaultTrainer
from detectron2.config import get_cfg

cfg = get_cfg()
cfg.merge_from_file(model_zoo.get_config_file("COCO-InstanceSegmentation/
→mask_rcnn_R_50_FPN_3x.yaml"))
cfg.DATASETS.TRAIN = ('balloon/train', )
cfg.DATASETS.TEST = ()
cfg.DATA_LOADER.NUM_WORKERS = 2
cfg.MODEL.WEIGHTS = "detectron2://COCO-InstanceSegmentation/
→mask_rcnn_R_50_FPN_3x/137849600/model_final_f10217.pkl"
cfg.SOLVER.IMS_PER_BATCH = 2
cfg.SOLVER.BASE_LR = 0.00025
cfg.SOLVER.MAX_ITER = 300
cfg.MODEL.ROI_HEADS.BATCH_SIZE_PER_IMAGE = 128
cfg.MODEL.ROI_HEADS.NUM_CLASSES = 1

```

```

os.makedirs(cfg.OUTPUT_DIR, exist_ok = True)
trainer = DefaultTrainer(cfg)
trainer.resume_or_load(resume = False)
trainer.train()

```

```

[10/16 18:22:32 d2.engine.defaults]: Model:
GeneralizedRCNN(
    (backbone): FPN(
        (fpn_lateral2): Conv2d(256, 256, kernel_size=(1, 1), stride=(1, 1))
        (fpn_output2): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
        (fpn_lateral3): Conv2d(512, 256, kernel_size=(1, 1), stride=(1, 1))
        (fpn_output3): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
        (fpn_lateral4): Conv2d(1024, 256, kernel_size=(1, 1), stride=(1, 1))
        (fpn_output4): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
        (fpn_lateral5): Conv2d(2048, 256, kernel_size=(1, 1), stride=(1, 1))
        (fpn_output5): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
        (top_block): LastLevelMaxPool()
    (bottom_up): ResNet(
        (stem): BasicStem(
            (conv1): Conv2d(
                3, 64, kernel_size=(7, 7), stride=(2, 2), padding=(3, 3), bias=False
                (norm): FrozenBatchNorm2d(num_features=64, eps=1e-05)
            )
        )
        (res2): Sequential(
            (0): BottleneckBlock(
                (shortcut): Conv2d(
                    64, 256, kernel_size=(1, 1), stride=(1, 1), bias=False
                    (norm): FrozenBatchNorm2d(num_features=256, eps=1e-05)
                )
                (conv1): Conv2d(
                    64, 64, kernel_size=(1, 1), stride=(1, 1), bias=False
                    (norm): FrozenBatchNorm2d(num_features=64, eps=1e-05)
                )
                (conv2): Conv2d(
                    64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
bias=False
                    (norm): FrozenBatchNorm2d(num_features=64, eps=1e-05)
                )
                (conv3): Conv2d(
                    64, 256, kernel_size=(1, 1), stride=(1, 1), bias=False
                    (norm): FrozenBatchNorm2d(num_features=256, eps=1e-05)
                )
            )
        )
    )
)
```

```

        )
    )
(1): BottleneckBlock(
    (conv1): Conv2d(
        256, 64, kernel_size=(1, 1), stride=(1, 1), bias=False
        (norm): FrozenBatchNorm2d(num_features=64, eps=1e-05)
    )
    (conv2): Conv2d(
        64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
bias=False
        (norm): FrozenBatchNorm2d(num_features=64, eps=1e-05)
    )
    (conv3): Conv2d(
        64, 256, kernel_size=(1, 1), stride=(1, 1), bias=False
        (norm): FrozenBatchNorm2d(num_features=256, eps=1e-05)
    )
)
(2): BottleneckBlock(
    (conv1): Conv2d(
        256, 64, kernel_size=(1, 1), stride=(1, 1), bias=False
        (norm): FrozenBatchNorm2d(num_features=64, eps=1e-05)
    )
    (conv2): Conv2d(
        64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
bias=False
        (norm): FrozenBatchNorm2d(num_features=64, eps=1e-05)
    )
    (conv3): Conv2d(
        64, 256, kernel_size=(1, 1), stride=(1, 1), bias=False
        (norm): FrozenBatchNorm2d(num_features=256, eps=1e-05)
    )
)
)
(res3): Sequential(
    (0): BottleneckBlock(
        (shortcut): Conv2d(
            256, 512, kernel_size=(1, 1), stride=(2, 2), bias=False
            (norm): FrozenBatchNorm2d(num_features=512, eps=1e-05)
        )
        (conv1): Conv2d(
            256, 128, kernel_size=(1, 1), stride=(2, 2), bias=False
            (norm): FrozenBatchNorm2d(num_features=128, eps=1e-05)
        )
        (conv2): Conv2d(
            128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
bias=False
            (norm): FrozenBatchNorm2d(num_features=128, eps=1e-05)
        )
    )
)

```

```

(conv3): Conv2d(
    128, 512, kernel_size=(1, 1), stride=(1, 1), bias=False
    (norm): FrozenBatchNorm2d(num_features=512, eps=1e-05)
)
)
(1): BottleneckBlock(
    (conv1): Conv2d(
        512, 128, kernel_size=(1, 1), stride=(1, 1), bias=False
        (norm): FrozenBatchNorm2d(num_features=128, eps=1e-05)
    )
    (conv2): Conv2d(
        128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
bias=False
        (norm): FrozenBatchNorm2d(num_features=128, eps=1e-05)
    )
    (conv3): Conv2d(
        128, 512, kernel_size=(1, 1), stride=(1, 1), bias=False
        (norm): FrozenBatchNorm2d(num_features=512, eps=1e-05)
    )
)
(2): BottleneckBlock(
    (conv1): Conv2d(
        512, 128, kernel_size=(1, 1), stride=(1, 1), bias=False
        (norm): FrozenBatchNorm2d(num_features=128, eps=1e-05)
    )
    (conv2): Conv2d(
        128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
bias=False
        (norm): FrozenBatchNorm2d(num_features=128, eps=1e-05)
    )
    (conv3): Conv2d(
        128, 512, kernel_size=(1, 1), stride=(1, 1), bias=False
        (norm): FrozenBatchNorm2d(num_features=512, eps=1e-05)
    )
)
(3): BottleneckBlock(
    (conv1): Conv2d(
        512, 128, kernel_size=(1, 1), stride=(1, 1), bias=False
        (norm): FrozenBatchNorm2d(num_features=128, eps=1e-05)
    )
    (conv2): Conv2d(
        128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
bias=False
        (norm): FrozenBatchNorm2d(num_features=128, eps=1e-05)
    )
    (conv3): Conv2d(
        128, 512, kernel_size=(1, 1), stride=(1, 1), bias=False
        (norm): FrozenBatchNorm2d(num_features=512, eps=1e-05)
    )
)

```

```

        )
    )
)
(res4): Sequential(
    (0): BottleneckBlock(
        (shortcut): Conv2d(
            512, 1024, kernel_size=(1, 1), stride=(2, 2), bias=False
            (norm): FrozenBatchNorm2d(num_features=1024, eps=1e-05)
        )
        (conv1): Conv2d(
            512, 256, kernel_size=(1, 1), stride=(2, 2), bias=False
            (norm): FrozenBatchNorm2d(num_features=256, eps=1e-05)
        )
        (conv2): Conv2d(
            256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
bias=False
            (norm): FrozenBatchNorm2d(num_features=256, eps=1e-05)
        )
        (conv3): Conv2d(
            256, 1024, kernel_size=(1, 1), stride=(1, 1), bias=False
            (norm): FrozenBatchNorm2d(num_features=1024, eps=1e-05)
        )
    )
    (1): BottleneckBlock(
        (conv1): Conv2d(
            1024, 256, kernel_size=(1, 1), stride=(1, 1), bias=False
            (norm): FrozenBatchNorm2d(num_features=256, eps=1e-05)
        )
        (conv2): Conv2d(
            256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
bias=False
            (norm): FrozenBatchNorm2d(num_features=256, eps=1e-05)
        )
        (conv3): Conv2d(
            256, 1024, kernel_size=(1, 1), stride=(1, 1), bias=False
            (norm): FrozenBatchNorm2d(num_features=1024, eps=1e-05)
        )
    )
    (2): BottleneckBlock(
        (conv1): Conv2d(
            1024, 256, kernel_size=(1, 1), stride=(1, 1), bias=False
            (norm): FrozenBatchNorm2d(num_features=256, eps=1e-05)
        )
        (conv2): Conv2d(
            256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
bias=False
            (norm): FrozenBatchNorm2d(num_features=256, eps=1e-05)
        )
    )
)

```

```

(conv3): Conv2d(
    256, 1024, kernel_size=(1, 1), stride=(1, 1), bias=False
    (norm): FrozenBatchNorm2d(num_features=1024, eps=1e-05)
)
)
(3): BottleneckBlock(
    (conv1): Conv2d(
        1024, 256, kernel_size=(1, 1), stride=(1, 1), bias=False
        (norm): FrozenBatchNorm2d(num_features=256, eps=1e-05)
    )
    (conv2): Conv2d(
        256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
bias=False
        (norm): FrozenBatchNorm2d(num_features=256, eps=1e-05)
    )
    (conv3): Conv2d(
        256, 1024, kernel_size=(1, 1), stride=(1, 1), bias=False
        (norm): FrozenBatchNorm2d(num_features=1024, eps=1e-05)
    )
)
(4): BottleneckBlock(
    (conv1): Conv2d(
        1024, 256, kernel_size=(1, 1), stride=(1, 1), bias=False
        (norm): FrozenBatchNorm2d(num_features=256, eps=1e-05)
    )
    (conv2): Conv2d(
        256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
bias=False
        (norm): FrozenBatchNorm2d(num_features=256, eps=1e-05)
    )
    (conv3): Conv2d(
        256, 1024, kernel_size=(1, 1), stride=(1, 1), bias=False
        (norm): FrozenBatchNorm2d(num_features=1024, eps=1e-05)
    )
)
(5): BottleneckBlock(
    (conv1): Conv2d(
        1024, 256, kernel_size=(1, 1), stride=(1, 1), bias=False
        (norm): FrozenBatchNorm2d(num_features=256, eps=1e-05)
    )
    (conv2): Conv2d(
        256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
bias=False
        (norm): FrozenBatchNorm2d(num_features=256, eps=1e-05)
    )
    (conv3): Conv2d(
        256, 1024, kernel_size=(1, 1), stride=(1, 1), bias=False
        (norm): FrozenBatchNorm2d(num_features=1024, eps=1e-05)
    )
)

```

```

        )
    )
)
(res5): Sequential(
    (0): BottleneckBlock(
        (shortcut): Conv2d(
            1024, 2048, kernel_size=(1, 1), stride=(2, 2), bias=False
            (norm): FrozenBatchNorm2d(num_features=2048, eps=1e-05)
        )
        (conv1): Conv2d(
            1024, 512, kernel_size=(1, 1), stride=(2, 2), bias=False
            (norm): FrozenBatchNorm2d(num_features=512, eps=1e-05)
        )
        (conv2): Conv2d(
            512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
bias=False
            (norm): FrozenBatchNorm2d(num_features=512, eps=1e-05)
        )
        (conv3): Conv2d(
            512, 2048, kernel_size=(1, 1), stride=(1, 1), bias=False
            (norm): FrozenBatchNorm2d(num_features=2048, eps=1e-05)
        )
    )
    (1): BottleneckBlock(
        (conv1): Conv2d(
            2048, 512, kernel_size=(1, 1), stride=(1, 1), bias=False
            (norm): FrozenBatchNorm2d(num_features=512, eps=1e-05)
        )
        (conv2): Conv2d(
            512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
bias=False
            (norm): FrozenBatchNorm2d(num_features=512, eps=1e-05)
        )
        (conv3): Conv2d(
            512, 2048, kernel_size=(1, 1), stride=(1, 1), bias=False
            (norm): FrozenBatchNorm2d(num_features=2048, eps=1e-05)
        )
    )
    (2): BottleneckBlock(
        (conv1): Conv2d(
            2048, 512, kernel_size=(1, 1), stride=(1, 1), bias=False
            (norm): FrozenBatchNorm2d(num_features=512, eps=1e-05)
        )
        (conv2): Conv2d(
            512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
bias=False
            (norm): FrozenBatchNorm2d(num_features=512, eps=1e-05)
        )
    )
)

```



```

        (3): ROIAlign(output_size=(14, 14), spatial_scale=0.03125,
sampling_ratio=0, aligned=True)
    )
)
(mask_head): MaskRCNNConvUpsampleHead(
    (mask_fcn1): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
    (mask_fcn2): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
    (mask_fcn3): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
    (mask_fcn4): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
    (deconv): ConvTranspose2d(256, 256, kernel_size=(2, 2), stride=(2, 2))
    (predictor): Conv2d(256, 1, kernel_size=(1, 1), stride=(1, 1))
)
)
)
[10/16 18:22:34 d2.data.build]: Removed 0 images with no usable
annotations. 61 images left.
[10/16 18:22:34 d2.data.build]: Distribution of instances among all 1
categories:
| category | #instances |
|:-----:|:-----|
| balloon | 255 |
|
[10/16 18:22:34 d2.data.common]: Serializing 61 elements to byte
tensors and concatenating them all ...
[10/16 18:22:34 d2.data.common]: Serialized dataset takes 0.17 MiB
[10/16 18:22:34 d2.data.dataset_mapper]: Augmentations used in
training: [ResizeShortestEdge(short_edge_length=(640, 672, 704, 736, 768, 800),
max_size=1333, sample_style='choice'), RandomFlip()]
[10/16 18:22:34 d2.data.build]: Using training sampler TrainingSampler

Skip loading parameter 'roi_heads.box_predictor.cls_score.weight' to the model
due to incompatible shapes: (81, 1024) in the checkpoint but (2, 1024) in the
model! You might want to double check if this is expected.
Skip loading parameter 'roi_heads.box_predictor.cls_score.bias' to the model due
to incompatible shapes: (81,) in the checkpoint but (2,) in the model! You might
want to double check if this is expected.
Skip loading parameter 'roi_heads.box_predictor.bbox_pred.weight' to the model
due to incompatible shapes: (320, 1024) in the checkpoint but (4, 1024) in the
model! You might want to double check if this is expected.
Skip loading parameter 'roi_heads.box_predictor.bbox_pred.bias' to the model due
to incompatible shapes: (320,) in the checkpoint but (4,) in the model! You
might want to double check if this is expected.

```

Skip loading parameter 'roi_heads.mask_head.predictor.weight' to the model due to incompatible shapes: (80, 256, 1, 1) in the checkpoint but (1, 256, 1, 1) in the model! You might want to double check if this is expected.

Skip loading parameter 'roi_heads.mask_head.predictor.bias' to the model due to incompatible shapes: (80,) in the checkpoint but (1,) in the model! You might want to double check if this is expected.

```
[10/16 18:22:39 d2.engine.train_loop]: Starting training from iteration 0
[10/16 18:22:52 d2.utils.events]: eta: 0:02:48 iter: 19 total_loss: 1.985 loss_cls: 0.633 loss_box_reg: 0.603 loss_mask: 0.680 loss_rpn_cls: 0.030 loss_rpn_loc: 0.009 time: 0.6013 data_time: 0.0301 lr: 0.000005 max_mem: 2863M
[10/16 18:23:04 d2.utils.events]: eta: 0:02:35 iter: 39 total_loss: 1.937 loss_cls: 0.589 loss_box_reg: 0.526 loss_mask: 0.654 loss_rpn_cls: 0.036 loss_rpn_loc: 0.007 time: 0.6012 data_time: 0.0070 lr: 0.000010 max_mem: 2863M
[10/16 18:23:16 d2.utils.events]: eta: 0:02:24 iter: 59 total_loss: 1.875 loss_cls: 0.537 loss_box_reg: 0.615 loss_mask: 0.594 loss_rpn_cls: 0.034 loss_rpn_loc: 0.007 time: 0.6022 data_time: 0.0073 lr: 0.000015 max_mem: 2863M
[10/16 18:23:28 d2.utils.events]: eta: 0:02:13 iter: 79 total_loss: 1.684 loss_cls: 0.487 loss_box_reg: 0.627 loss_mask: 0.533 loss_rpn_cls: 0.035 loss_rpn_loc: 0.008 time: 0.6066 data_time: 0.0081 lr: 0.000020 max_mem: 2863M
[10/16 18:23:41 d2.utils.events]: eta: 0:02:01 iter: 99 total_loss: 1.636 loss_cls: 0.428 loss_box_reg: 0.745 loss_mask: 0.462 loss_rpn_cls: 0.028 loss_rpn_loc: 0.008 time: 0.6082 data_time: 0.0077 lr: 0.000025 max_mem: 2863M
[10/16 18:23:53 d2.utils.events]: eta: 0:01:50 iter: 119 total_loss: 1.504 loss_cls: 0.381 loss_box_reg: 0.662 loss_mask: 0.395 loss_rpn_cls: 0.022 loss_rpn_loc: 0.006 time: 0.6119 data_time: 0.0148 lr: 0.000030 max_mem: 2863M
[10/16 18:24:06 d2.utils.events]: eta: 0:01:38 iter: 139 total_loss: 1.430 loss_cls: 0.367 loss_box_reg: 0.672 loss_mask: 0.376 loss_rpn_cls: 0.029 loss_rpn_loc: 0.009 time: 0.6149 data_time: 0.0069 lr: 0.000035 max_mem: 2863M
[10/16 18:24:19 d2.utils.events]: eta: 0:01:26 iter: 159 total_loss: 1.351 loss_cls: 0.306 loss_box_reg: 0.640 loss_mask: 0.311 loss_rpn_cls: 0.025 loss_rpn_loc: 0.009 time: 0.6179 data_time: 0.0073 lr: 0.000040 max_mem: 2863M
[10/16 18:24:31 d2.utils.events]: eta: 0:01:14 iter: 179 total_loss: 1.251 loss_cls: 0.278 loss_box_reg: 0.610 loss_mask: 0.283 loss_rpn_cls: 0.027 loss_rpn_loc: 0.008 time: 0.6172 data_time: 0.0089 lr: 0.000045 max_mem: 2863M
[10/16 18:24:44 d2.utils.events]: eta: 0:01:02 iter: 199 total_loss: 1.173 loss_cls: 0.251 loss_box_reg: 0.676 loss_mask: 0.254 loss_rpn_cls: 0.013 loss_rpn_loc: 0.008 time: 0.6196 data_time: 0.0075 lr: 0.000050
```

```

max_mem: 2863M
[10/16 18:24:56 d2.utils.events]: eta: 0:00:49 iter: 219 total_loss:
1.138 loss_cls: 0.225 loss_box_reg: 0.636 loss_mask: 0.206 loss_rpn_cls:
0.022 loss_rpn_loc: 0.013 time: 0.6193 data_time: 0.0084 lr: 0.000055
max_mem: 2863M
[10/16 18:25:09 d2.utils.events]: eta: 0:00:37 iter: 239 total_loss:
1.073 loss_cls: 0.179 loss_box_reg: 0.636 loss_mask: 0.170 loss_rpn_cls:
0.026 loss_rpn_loc: 0.006 time: 0.6219 data_time: 0.0076 lr: 0.000060
max_mem: 2863M
[10/16 18:25:22 d2.utils.events]: eta: 0:00:25 iter: 259 total_loss:
1.001 loss_cls: 0.174 loss_box_reg: 0.583 loss_mask: 0.175 loss_rpn_cls:
0.017 loss_rpn_loc: 0.008 time: 0.6233 data_time: 0.0086 lr: 0.000065
max_mem: 2863M
[10/16 18:25:35 d2.utils.events]: eta: 0:00:13 iter: 279 total_loss:
0.925 loss_cls: 0.156 loss_box_reg: 0.588 loss_mask: 0.161 loss_rpn_cls:
0.019 loss_rpn_loc: 0.007 time: 0.6253 data_time: 0.0070 lr: 0.000070
max_mem: 2863M
[10/16 18:25:50 d2.utils.events]: eta: 0:00:00 iter: 299 total_loss:
0.761 loss_cls: 0.124 loss_box_reg: 0.472 loss_mask: 0.156 loss_rpn_cls:
0.016 loss_rpn_loc: 0.007 time: 0.6272 data_time: 0.0117 lr: 0.000075
max_mem: 2863M
[10/16 18:25:50 d2.engine.hooks]: Overall training speed: 297
iterations in 0:03:06 (0.6293 s / it)
[10/16 18:25:50 d2.engine.hooks]: Total training time: 0:03:09 (0:00:02
on hooks)

```

```

[ ]: from detectron2.utils.visualizer import ColorMode

cfg.MODEL.WEIGHTS = os.path.join(cfg.OUTPUT_DIR, 'model_final.pth')
cfg.MODEL.ROI_HEADS.SCORE_THRESH_TEST = 0.498

predictor = DefaultPredictor(cfg)
outputs = predictor(im)

v = Visualizer(im[:, :, ::-1], metadata = balloon_metadata, scale=1.2, →
instance_mode = ColorMode.IMAGE_BW)
v = v.draw_instance_predictions(outputs["instances"].to("cpu"))
cv2_imshow(v.get_image()[:, :, ::-1])

```

