Transfer learning - Hat

Pre-trained weight of coco model

Liu Yen-Fu and Mattia Cerino - 24/NOV/2019 Group25

First- Model Follow the balloon example

Using TensorFlow backend.

```
Weights:  coco
Dataset:  /content/Mask_RCNN/tmp/balloon/
Logs:  /content/Mask_RCNN/logs
Configurations:
BACKBONE                       resnet101
BACKBONE_STRIDES               [4, 8, 16, 32, 64]
BATCH_SIZE                     2
BBOX_STD_DEV                   [0.1 0.1 0.2 0.2]
COMPUTE_BACKBONE_SHAPE         None
DETECTION_MAX_INSTANCES        100
DETECTION_MIN_CONFIDENCE       0.9
DETECTION_NMS_THRESHOLD        0.3
FPN_CLASSIF_FC_LAYERS_SIZE     1024
GPU_COUNT                      1
GRADIENT_CLIP_NORM             5.0
IMAGES_PER_GPU                 2
IMAGE_CHANNEL_COUNT            3
IMAGE_MAX_DIM                  1024
IMAGE_META_SIZE                14
IMAGE_MIN_DIM                  800
IMAGE_MIN_SCALE                0
IMAGE_RESIZE_MODE              square
IMAGE_SHAPE                    [1024 1024    3]
LEARNING_MOMENTUM              0.9
LEARNING_RATE                  0.001
LOSS_WEIGHTS                   {'rpn_class_loss': 1.0, 'rpn_bbox_loss': 1.0,
'mrcnn_class_loss': 1.0, 'mrcnn_bbox_loss': 1.0, 'mrcnn_mask_loss': 1.0}
MASK_POOL_SIZE                 14
MASK_SHAPE                     [28, 28]
MAX_GT_INSTANCES               100
MEAN_PIXEL                     [123.7 116.8 103.9]
MINI_MASK_SHAPE                (56, 56)
NAME                           balloon
NUM_CLASSES                    2
POOL_SIZE                      7
POST_NMS_ROIS_INFERENCE        1000
POST_NMS_ROIS_TRAINING         2000
PRE_NMS_LIMIT                  6000
ROI_POSITIVE_RATIO             0.33
RPN_ANCHOR_RATIOS              [0.5, 1, 2]
RPN_ANCHOR_SCALES              (32, 64, 128, 256, 512)
RPN_ANCHOR_STRIDE              1
RPN_BBOX_STD_DEV               [0.1 0.1 0.2 0.2]
RPN_NMS_THRESHOLD              0.7
RPN_TRAIN_ANCHORS_PER_IMAGE    256
STEPS_PER_EPOCH                100
TOP_DOWN_PYRAMID_SIZE          256
TRAIN_BN                       False
TRAIN_ROIS_PER_IMAGE           200
USE_MINI_MASK                  True
USE_RPN_ROIS                   True
VALIDATION_STEPS               50
WEIGHT_DECAY                   0.0001
```

```
Training network heads

Starting at epoch 0. LR=0.001

Checkpoint Path: /content/Mask_RCNN/logs/balloon20191113T2117/
mask_rcnn_balloon_{epoch:04d}.h5
Selecting layers to train
fpn_c5p5                (Conv2D)
fpn_c4p4                (Conv2D)
fpn_c3p3                (Conv2D)
fpn_c2p2                (Conv2D)
fpn_p5                  (Conv2D)
fpn_p2                  (Conv2D)
fpn_p3                  (Conv2D)
fpn_p4                  (Conv2D)
In model:  rpn_model
    rpn_conv_shared         (Conv2D)
    rpn_class_raw           (Conv2D)
    rpn_bbox_pred           (Conv2D)
mrcnn_mask_conv1        (TimeDistributed)
mrcnn_mask_bn1          (TimeDistributed)
mrcnn_mask_conv2        (TimeDistributed)
mrcnn_mask_bn2          (TimeDistributed)
mrcnn_class_conv1       (TimeDistributed)
mrcnn_class_bn1         (TimeDistributed)
mrcnn_mask_conv3        (TimeDistributed)
mrcnn_mask_bn3          (TimeDistributed)
mrcnn_class_conv2       (TimeDistributed)
mrcnn_class_bn2         (TimeDistributed)
mrcnn_mask_conv4        (TimeDistributed)
mrcnn_mask_bn4          (TimeDistributed)
mrcnn_bbox_fc           (TimeDistributed)
mrcnn_mask_deconv       (TimeDistributed)
mrcnn_class_logits      (TimeDistributed)
mrcnn_mask              (TimeDistributed)
```

```
100/100 [==============================] - 353s 4s/step - loss: 0.9005 - rpn_class_loss: 0.0284 -
rpn_bbox_loss: 0.1759 - mrcnn_class_loss: 0.1016 - mrcnn_bbox_loss: 0.4059 - mrcnn_mask_loss:
0.1887 - val_loss: 0.8265 - val_rpn_class_loss: 0.0381 - val_rpn_bbox_loss: 0.3592 -
val_mrcnn_class_loss: 0.0731 - val_mrcnn_bbox_loss: 0.2129 - val_mrcnn_mask_loss: 0.1431
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/callbacks.py:1265: The name
tf.Summary is deprecated. Please use tf.compat.v1.Summary instead.

Epoch 2/30
100/100 [==============================] - 284s 3s/step - loss: 0.5199 - rpn_class_loss: 0.0159 -
rpn_bbox_loss: 0.1138 - mrcnn_class_loss: 0.0726 - mrcnn_bbox_loss: 0.1723 - mrcnn_mask_loss:
0.1453 - val_loss: 0.7028 - val_rpn_class_loss: 0.0358 - val_rpn_bbox_loss: 0.3306 -
val_mrcnn_class_loss: 0.0629 - val_mrcnn_bbox_loss: 0.1602 - val_mrcnn_mask_loss: 0.1133
Epoch 3/30
100/100 [==============================] - 284s 3s/step - loss: 0.3503 - rpn_class_loss: 0.0095 -
rpn_bbox_loss: 0.0663 - mrcnn_class_loss: 0.0477 - mrcnn_bbox_loss: 0.1112 - mrcnn_mask_loss:
0.1155 - val_loss: 0.7430 - val_rpn_class_loss: 0.0476 - val_rpn_bbox_loss: 0.3730 -
val_mrcnn_class_loss: 0.0607 - val_mrcnn_bbox_loss: 0.1489 - val_mrcnn_mask_loss: 0.1127
Epoch 4/30
100/100 [==============================] - 283s 3s/step - loss: 0.2988 - rpn_class_loss: 0.0079 -
rpn_bbox_loss: 0.0533 - mrcnn_class_loss: 0.0382 - mrcnn_bbox_loss: 0.0933 - mrcnn_mask_loss:
0.1062 - val_loss: 0.7620 - val_rpn_class_loss: 0.0425 - val_rpn_bbox_loss: 0.3932 -
val_mrcnn_class_loss: 0.0730 - val_mrcnn_bbox_loss: 0.1440 - val_mrcnn_mask_loss: 0.1092
Epoch 5/30
100/100 [==============================] - 284s 3s/step - loss: 0.2641 - rpn_class_loss: 0.0071 -
rpn_bbox_loss: 0.0489 - mrcnn_class_loss: 0.0322 - mrcnn_bbox_loss: 0.0748 - mrcnn_mask_loss:
0.1011 - val_loss: 0.7078 - val_rpn_class_loss: 0.0436 - val_rpn_bbox_loss: 0.3735 -
val_mrcnn_class_loss: 0.0620 - val_mrcnn_bbox_loss: 0.1209 - val_mrcnn_mask_loss: 0.1078
Epoch 6/30
100/100 [==============================] - 284s 3s/step - loss: 0.2249 - rpn_class_loss: 0.0061 -
rpn_bbox_loss: 0.0424 - mrcnn_class_loss: 0.0297 - mrcnn_bbox_loss: 0.0508 - mrcnn_mask_loss:
0.0958 - val_loss: 0.9233 - val_rpn_class_loss: 0.0564 - val_rpn_bbox_loss: 0.5668 -
val_mrcnn_class_loss: 0.0566 - val_mrcnn_bbox_loss: 0.1244 - val_mrcnn_mask_loss: 0.1191
Epoch 7/30
100/100 [==============================] - 283s 3s/step - loss: 0.2026 - rpn_class_loss: 0.0045 -
rpn_bbox_loss: 0.0369 - mrcnn_class_loss: 0.0285 - mrcnn_bbox_loss: 0.0455 - mrcnn_mask_loss:
0.0872 - val_loss: 0.8708 - val_rpn_class_loss: 0.0606 - val_rpn_bbox_loss: 0.5146 -
val_mrcnn_class_loss: 0.0605 - val_mrcnn_bbox_loss: 0.1226 - val_mrcnn_mask_loss: 0.1126
Epoch 8/30
100/100 [==============================] - 283s 3s/step - loss: 0.1917 - rpn_class_loss: 0.0039 -
rpn_bbox_loss: 0.0363 - mrcnn_class_loss: 0.0226 - mrcnn_bbox_loss: 0.0403 - mrcnn_mask_loss:
0.0886 - val_loss: 0.8891 - val_rpn_class_loss: 0.0573 - val_rpn_bbox_loss: 0.5483 -
val_mrcnn_class_loss: 0.0707 - val_mrcnn_bbox_loss: 0.0983 - val_mrcnn_mask_loss: 0.1145
Epoch 9/30
100/100 [==============================] - 287s 3s/step - loss: 0.1808 - rpn_class_loss: 0.0034 -
rpn_bbox_loss: 0.0327 - mrcnn_class_loss: 0.0215 - mrcnn_bbox_loss: 0.0394 - mrcnn_mask_loss:
0.0839 - val_loss: 0.9122 - val_rpn_class_loss: 0.0647 - val_rpn_bbox_loss: 0.5664 -
val_mrcnn_class_loss: 0.0579 - val_mrcnn_bbox_loss: 0.1097 - val_mrcnn_mask_loss: 0.1136
Epoch 10/30
100/100 [==============================] - 289s 3s/step - loss: 0.1650 - rpn_class_loss: 0.0033 -
rpn_bbox_loss: 0.0258 - mrcnn_class_loss: 0.0210 - mrcnn_bbox_loss: 0.0322 - mrcnn_mask_loss:
0.0827 - val_loss: 0.9272 - val_rpn_class_loss: 0.0709 - val_rpn_bbox_loss: 0.5216 -
val_mrcnn_class_loss: 0.1036 - val_mrcnn_bbox_loss: 0.1059 - val_mrcnn_mask_loss: 0.1251
Epoch 11/30
100/100 [==============================] - 289s 3s/step - loss: 0.1709 - rpn_class_loss: 0.0035 -
rpn_bbox_loss: 0.0369 - mrcnn_class_loss: 0.0208 - mrcnn_bbox_loss: 0.0334 - mrcnn_mask_loss:
0.0763 - val_loss: 0.8623 - val_rpn_class_loss: 0.0649 - val_rpn_bbox_loss: 0.5328 -
val_mrcnn_class_loss: 0.0612 - val_mrcnn_bbox_loss: 0.0978 - val_mrcnn_mask_loss: 0.1055
Epoch 12/30
100/100 [==============================] - 288s 3s/step - loss: 0.1384 - rpn_class_loss: 0.0026 -
rpn_bbox_loss: 0.0229 - mrcnn_class_loss: 0.0160 - mrcnn_bbox_loss: 0.0205 - mrcnn_mask_loss:
0.0765 - val_loss: 0.7997 - val_rpn_class_loss: 0.0574 - val_rpn_bbox_loss: 0.4305 -
val_mrcnn_class_loss: 0.0839 - val_mrcnn_bbox_loss: 0.1121 - val_mrcnn_mask_loss: 0.1158
Epoch 13/30
100/100 [==============================] - 290s 3s/step - loss: 0.1264 - rpn_class_loss: 0.0030 -
rpn_bbox_loss: 0.0170 - mrcnn_class_loss: 0.0197 - mrcnn_bbox_loss: 0.0159 - mrcnn_mask_loss:
0.0708 - val_loss: 0.9300 - val_rpn_class_loss: 0.0799 - val_rpn_bbox_loss: 0.5264 -
val_mrcnn_class_loss: 0.0977 - val_mrcnn_bbox_loss: 0.1075 - val_mrcnn_mask_loss: 0.1185
Epoch 14/30
100/100 [==============================] - 287s 3s/step - loss: 0.1256 - rpn_class_loss: 0.0029 -
rpn_bbox_loss: 0.0199 - mrcnn_class_loss: 0.0173 - mrcnn_bbox_loss: 0.0182 - mrcnn_mask_loss:
```

```
0.0673 - val_loss: 0.8404 - val_rpn_class_loss: 0.0702 - val_rpn_bbox_loss: 0.4953 -
val_mrcnn_class_loss: 0.0740 - val_mrcnn_bbox_loss: 0.0886 - val_mrcnn_mask_loss: 0.1123
Epoch 15/30
100/100 [==============================] - 288s 3s/step - loss: 0.1177 - rpn_class_loss: 0.0024 -
rpn_bbox_loss: 0.0172 - mrcnn_class_loss: 0.0161 - mrcnn_bbox_loss: 0.0151 - mrcnn_mask_loss:
0.0669 - val_loss: 0.9119 - val_rpn_class_loss: 0.0732 - val_rpn_bbox_loss: 0.5548 -
val_mrcnn_class_loss: 0.0749 - val_mrcnn_bbox_loss: 0.0992 - val_mrcnn_mask_loss: 0.1097
Epoch 16/30
100/100 [==============================] - 288s 3s/step - loss: 0.1149 - rpn_class_loss: 0.0025 -
rpn_bbox_loss: 0.0136 - mrcnn_class_loss: 0.0161 - mrcnn_bbox_loss: 0.0154 - mrcnn_mask_loss:
0.0674 - val_loss: 0.9541 - val_rpn_class_loss: 0.0770 - val_rpn_bbox_loss: 0.5497 -
val_mrcnn_class_loss: 0.0841 - val_mrcnn_bbox_loss: 0.1135 - val_mrcnn_mask_loss: 0.1298
Epoch 17/30
100/100 [==============================] - 288s 3s/step - loss: 0.1178 - rpn_class_loss: 0.0023 -
rpn_bbox_loss: 0.0186 - mrcnn_class_loss: 0.0154 - mrcnn_bbox_loss: 0.0181 - mrcnn_mask_loss:
0.0633 - val_loss: 0.7940 - val_rpn_class_loss: 0.0704 - val_rpn_bbox_loss: 0.4215 -
val_mrcnn_class_loss: 0.0929 - val_mrcnn_bbox_loss: 0.0909 - val_mrcnn_mask_loss: 0.1183
Epoch 18/30
100/100 [==============================] - 287s 3s/step - loss: 0.1169 - rpn_class_loss: 0.0026 -
rpn_bbox_loss: 0.0186 - mrcnn_class_loss: 0.0160 - mrcnn_bbox_loss: 0.0167 - mrcnn_mask_loss:
0.0630 - val_loss: 0.8518 - val_rpn_class_loss: 0.0815 - val_rpn_bbox_loss: 0.4699 -
val_mrcnn_class_loss: 0.1003 - val_mrcnn_bbox_loss: 0.0875 - val_mrcnn_mask_loss: 0.1126
Epoch 19/30
100/100 [==============================] - 289s 3s/step - loss: 0.0971 - rpn_class_loss: 0.0025 -
rpn_bbox_loss: 0.0122 - mrcnn_class_loss: 0.0151 - mrcnn_bbox_loss: 0.0117 - mrcnn_mask_loss:
0.0556 - val_loss: 0.8155 - val_rpn_class_loss: 0.0759 - val_rpn_bbox_loss: 0.4544 -
val_mrcnn_class_loss: 0.0751 - val_mrcnn_bbox_loss: 0.0896 - val_mrcnn_mask_loss: 0.1205
Epoch 20/30
100/100 [==============================] - 288s 3s/step - loss: 0.0956 - rpn_class_loss: 0.0019 -
rpn_bbox_loss: 0.0109 - mrcnn_class_loss: 0.0128 - mrcnn_bbox_loss: 0.0117 - mrcnn_mask_loss:
0.0582 - val_loss: 0.9135 - val_rpn_class_loss: 0.0803 - val_rpn_bbox_loss: 0.5121 -
val_mrcnn_class_loss: 0.0970 - val_mrcnn_bbox_loss: 0.0983 - val_mrcnn_mask_loss: 0.1258
Epoch 21/30
100/100 [==============================] - 289s 3s/step - loss: 0.1038 - rpn_class_loss: 0.0019 -
rpn_bbox_loss: 0.0126 - mrcnn_class_loss: 0.0127 - mrcnn_bbox_loss: 0.0155 - mrcnn_mask_loss:
0.0612 - val_loss: 0.8896 - val_rpn_class_loss: 0.0842 - val_rpn_bbox_loss: 0.5007 -
val_mrcnn_class_loss: 0.0905 - val_mrcnn_bbox_loss: 0.0897 - val_mrcnn_mask_loss: 0.1245
Epoch 22/30
100/100 [==============================] - 289s 3s/step - loss: 0.0964 - rpn_class_loss: 0.0021 -
rpn_bbox_loss: 0.0113 - mrcnn_class_loss: 0.0138 - mrcnn_bbox_loss: 0.0126 - mrcnn_mask_loss:
0.0566 - val_loss: 0.8783 - val_rpn_class_loss: 0.0861 - val_rpn_bbox_loss: 0.4633 -
val_mrcnn_class_loss: 0.0953 - val_mrcnn_bbox_loss: 0.1079 - val_mrcnn_mask_loss: 0.1256
Epoch 23/30
100/100 [==============================] - 290s 3s/step - loss: 0.0860 - rpn_class_loss: 0.0018 -
rpn_bbox_loss: 0.0100 - mrcnn_class_loss: 0.0117 - mrcnn_bbox_loss: 0.0095 - mrcnn_mask_loss:
0.0531 - val_loss: 0.8459 - val_rpn_class_loss: 0.0864 - val_rpn_bbox_loss: 0.4587 -
val_mrcnn_class_loss: 0.0849 - val_mrcnn_bbox_loss: 0.0940 - val_mrcnn_mask_loss: 0.1219
Epoch 24/30
100/100 [==============================] - 287s 3s/step - loss: 0.0808 - rpn_class_loss: 0.0017 -
rpn_bbox_loss: 0.0071 - mrcnn_class_loss: 0.0126 - mrcnn_bbox_loss: 0.0070 - mrcnn_mask_loss:
0.0524 - val_loss: 0.9000 - val_rpn_class_loss: 0.0902 - val_rpn_bbox_loss: 0.4698 -
val_mrcnn_class_loss: 0.1086 - val_mrcnn_bbox_loss: 0.0974 - val_mrcnn_mask_loss: 0.1340
Epoch 25/30
100/100 [==============================] - 288s 3s/step - loss: 0.0794 - rpn_class_loss: 0.0014 -
rpn_bbox_loss: 0.0053 - mrcnn_class_loss: 0.0114 - mrcnn_bbox_loss: 0.0076 - mrcnn_mask_loss:
0.0536 - val_loss: 0.8436 - val_rpn_class_loss: 0.0915 - val_rpn_bbox_loss: 0.4616 -
val_mrcnn_class_loss: 0.0846 - val_mrcnn_bbox_loss: 0.0876 - val_mrcnn_mask_loss: 0.1184
Epoch 26/30
100/100 [==============================] - 285s 3s/step - loss: 0.0778 - rpn_class_loss: 0.0012 -
rpn_bbox_loss: 0.0064 - mrcnn_class_loss: 0.0099 - mrcnn_bbox_loss: 0.0069 - mrcnn_mask_loss:
0.0532 - val_loss: 0.9132 - val_rpn_class_loss: 0.0971 - val_rpn_bbox_loss: 0.4850 -
val_mrcnn_class_loss: 0.0997 - val_mrcnn_bbox_loss: 0.0960 - val_mrcnn_mask_loss: 0.1354
Epoch 27/30
100/100 [==============================] - 283s 3s/step - loss: 0.0866 - rpn_class_loss: 0.0014 -
rpn_bbox_loss: 0.0110 - mrcnn_class_loss: 0.0107 - mrcnn_bbox_loss: 0.0097 - mrcnn_mask_loss:
0.0538 - val_loss: 0.8864 - val_rpn_class_loss: 0.0942 - val_rpn_bbox_loss: 0.5046 -
val_mrcnn_class_loss: 0.0800 - val_mrcnn_bbox_loss: 0.0820 - val_mrcnn_mask_loss: 0.1256
Epoch 28/30
100/100 [==============================] - 283s 3s/step - loss: 0.0860 - rpn_class_loss: 0.0017 -
rpn_bbox_loss: 0.0117 - mrcnn_class_loss: 0.0109 - mrcnn_bbox_loss: 0.0090 - mrcnn_mask_loss:
```

```
0.0526 - val_loss: 0.9951 - val_rpn_class_loss: 0.1091 - val_rpn_bbox_loss: 0.5657 -
val_mrcnn_class_loss: 0.0909 - val_mrcnn_bbox_loss: 0.0920 - val_mrcnn_mask_loss: 0.1374
Epoch 29/30
100/100 [==============================] - 282s 3s/step - loss: 0.0766 - rpn_class_loss: 0.0016 -
rpn_bbox_loss: 0.0065 - mrcnn_class_loss: 0.0100 - mrcnn_bbox_loss: 0.0079 - mrcnn_mask_loss:
0.0507 - val_loss: 0.9595 - val_rpn_class_loss: 0.0986 - val_rpn_bbox_loss: 0.5200 -
val_mrcnn_class_loss: 0.1120 - val_mrcnn_bbox_loss: 0.0950 - val_mrcnn_mask_loss: 0.1339
Epoch 30/30
```

Hat- train with coco pre-trained weight

```
Using TensorFlow backend.
Downloading pretrained model to /content/Mask_RCNN/coco_model/samples/mask_rcnn_coco.h5 ...
... done downloading pretrained model!
Weights:  coco
Dataset:  /content/Mask_RCNN/coco_model/hat_dataset/
Logs:  /content/Mask_RCNN/coco_model/samples/logs
Configurations:
BACKBONE                       resnet101
BACKBONE_STRIDES               [4, 8, 16, 32, 64]
BATCH_SIZE                     1
BBOX_STD_DEV                   [0.1 0.1 0.2 0.2]
COMPUTE_BACKBONE_SHAPE         None
DETECTION_MAX_INSTANCES        100
DETECTION_MIN_CONFIDENCE       0.9
DETECTION_NMS_THRESHOLD        0.3
FPN_CLASSIF_FC_LAYERS_SIZE     1024
GPU_COUNT                      1
GRADIENT_CLIP_NORM             5.0
IMAGES_PER_GPU                 1
IMAGE_CHANNEL_COUNT            3
IMAGE_MAX_DIM                  1024
IMAGE_META_SIZE                14
IMAGE_MIN_DIM                  800
IMAGE_MIN_SCALE                0
IMAGE_RESIZE_MODE              square
IMAGE_SHAPE                    [1024 1024    3]
LEARNING_MOMENTUM              0.9
LEARNING_RATE                  0.001
LOSS_WEIGHTS                   {'rpn_class_loss': 1.0, 'rpn_bbox_loss': 1.0, 'mrcnn_class_loss':
1.0, 'mrcnn_bbox_loss': 1.0, 'mrcnn_mask_loss': 1.0}
MASK_POOL_SIZE                 14
MASK_SHAPE                     [28, 28]
MAX_GT_INSTANCES               100
MEAN_PIXEL                     [123.7 116.8 103.9]
MINI_MASK_SHAPE                (56, 56)
NAME                           hat
NUM_CLASSES                    2
POOL_SIZE                      7
POST_NMS_ROIS_INFERENCE        1000
POST_NMS_ROIS_TRAINING         2000
PRE_NMS_LIMIT                  6000
ROI_POSITIVE_RATIO             0.33
RPN_ANCHOR_RATIOS              [0.5, 1, 2]
RPN_ANCHOR_SCALES              (32, 64, 128, 256, 512)
RPN_ANCHOR_STRIDE              1
RPN_BBOX_STD_DEV               [0.1 0.1 0.2 0.2]
RPN_NMS_THRESHOLD              0.7
RPN_TRAIN_ANCHORS_PER_IMAGE    256
STEPS_PER_EPOCH                100
TOP_DOWN_PYRAMID_SIZE          256
TRAIN_BN                       False
TRAIN_ROIS_PER_IMAGE           200
USE_MINI_MASK                  True
USE_RPN_ROIS                   True
VALIDATION_STEPS               50
WEIGHT_DECAY                   0.0001
```

```
Training network heads

Starting at epoch 0. LR=0.001

Checkpoint Path: /content/Mask_RCNN/coco_model/samples/logs/hat20191114T1238/mask_rcnn_hat_{epoch:
04d}.h5
Selecting layers to train
fpn_c5p5               (Conv2D)
fpn_c4p4               (Conv2D)
fpn_c3p3               (Conv2D)
fpn_c2p2               (Conv2D)
fpn_p5                 (Conv2D)
fpn_p2                 (Conv2D)
fpn_p3                 (Conv2D)
fpn_p4                 (Conv2D)
In model:  rpn_model
    rpn_conv_shared        (Conv2D)
    rpn_class_raw          (Conv2D)
    rpn_bbox_pred          (Conv2D)
mrcnn_mask_conv1       (TimeDistributed)
mrcnn_mask_bn1         (TimeDistributed)
mrcnn_mask_conv2       (TimeDistributed)
mrcnn_mask_bn2         (TimeDistributed)
mrcnn_class_conv1      (TimeDistributed)
mrcnn_class_bn1        (TimeDistributed)
mrcnn_mask_conv3       (TimeDistributed)
mrcnn_mask_bn3         (TimeDistributed)
mrcnn_class_conv2      (TimeDistributed)
mrcnn_class_bn2        (TimeDistributed)
mrcnn_mask_conv4       (TimeDistributed)
mrcnn_mask_bn4         (TimeDistributed)
mrcnn_bbox_fc          (TimeDistributed)
mrcnn_mask_deconv      (TimeDistributed)
mrcnn_class_logits     (TimeDistributed)
mrcnn_mask             (TimeDistributed)
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/optimizers.py:793: The name
tf.train.Optimizer is deprecated. Please use tf.compat.v1.train.Optimizer instead.

/usr/local/lib/python3.6/dist-packages/tensorflow_core/python/framework/indexed_slices.py:424:
UserWarning: Converting sparse IndexedSlices to a dense Tensor of unknown shape. This may consume a
large amount of memory.
  "Converting sparse IndexedSlices to a dense Tensor of unknown shape. "
/usr/local/lib/python3.6/dist-packages/tensorflow_core/python/framework/indexed_slices.py:424:
UserWarning: Converting sparse IndexedSlices to a dense Tensor of unknown shape. This may consume a
large amount of memory.
  "Converting sparse IndexedSlices to a dense Tensor of unknown shape. "
/usr/local/lib/python3.6/dist-packages/tensorflow_core/python/framework/indexed_slices.py:424:
UserWarning: Converting sparse IndexedSlices to a dense Tensor of unknown shape. This may consume a
large amount of memory.
  "Converting sparse IndexedSlices to a dense Tensor of unknown shape. "
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow_backend.py:
1033: The name tf.assign_add is deprecated. Please use tf.compat.v1.assign_add instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow_backend.py:
1020: The name tf.assign is deprecated. Please use tf.compat.v1.assign instead.

/usr/local/lib/python3.6/dist-packages/keras/engine/training_generator.py:49: UserWarning: Using a
generator with `use_multiprocessing=True` and multiple workers may duplicate your data. Please
consider using the `keras.utils.Sequence class.
  UserWarning('Using a generator with `use_multiprocessing=True`'
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/callbacks.py:1122: The name
tf.summary.merge_all is deprecated. Please use tf.compat.v1.summary.merge_all instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/callbacks.py:1125: The name
tf.summary.FileWriter is deprecated. Please use tf.compat.v1.summary.FileWriter instead.

Epoch 1/30
2019-11-14 12:39:32.502306: I tensorflow/stream_executor/platform/default/dso_loader.cc:44]
Successfully opened dynamic library libcublas.so.10.0
```

```
2019-11-14 12:39:33.824645: I tensorflow/stream_executor/platform/default/dso_loader.cc:44]
Successfully opened dynamic library libcudnn.so.7
100/100 [==============================] - 203s 2s/step - loss: 1.5495 - rpn_class_loss: 0.0103 -
rpn_bbox_loss: 0.4105 - mrcnn_class_loss: 0.0974 - mrcnn_bbox_loss: 0.7093 - mrcnn_mask_loss:
0.3221 - val_loss: 1.1839 - val_rpn_class_loss: 0.0184 - val_rpn_bbox_loss: 0.3003 -
val_mrcnn_class_loss: 0.0612 - val_mrcnn_bbox_loss: 0.5289 - val_mrcnn_mask_loss: 0.2750
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/callbacks.py:1265: The name
tf.Summary is deprecated. Please use tf.compat.v1.Summary instead.

Epoch 2/30
100/100 [==============================] - 149s 1s/step - loss: 0.7499 - rpn_class_loss: 0.0055 -
rpn_bbox_loss: 0.2378 - mrcnn_class_loss: 0.0314 - mrcnn_bbox_loss: 0.3045 - mrcnn_mask_loss:
0.1707 - val_loss: 1.0167 - val_rpn_class_loss: 0.0181 - val_rpn_bbox_loss: 0.3426 -
val_mrcnn_class_loss: 0.0511 - val_mrcnn_bbox_loss: 0.3535 - val_mrcnn_mask_loss: 0.2514
Epoch 3/30
100/100 [==============================] - 148s 1s/step - loss: 0.6142 - rpn_class_loss: 0.0047 -
rpn_bbox_loss: 0.2109 - mrcnn_class_loss: 0.0261 - mrcnn_bbox_loss: 0.2120 - mrcnn_mask_loss:
0.1605 - val_loss: 1.2929 - val_rpn_class_loss: 0.0258 - val_rpn_bbox_loss: 0.4021 -
val_mrcnn_class_loss: 0.0424 - val_mrcnn_bbox_loss: 0.5314 - val_mrcnn_mask_loss: 0.2912
Epoch 4/30
100/100 [==============================] - 148s 1s/step - loss: 0.4184 - rpn_class_loss: 0.0049 -
rpn_bbox_loss: 0.1604 - mrcnn_class_loss: 0.0174 - mrcnn_bbox_loss: 0.1216 - mrcnn_mask_loss:
0.1142 - val_loss: 0.9857 - val_rpn_class_loss: 0.0147 - val_rpn_bbox_loss: 0.3411 -
val_mrcnn_class_loss: 0.0271 - val_mrcnn_bbox_loss: 0.3629 - val_mrcnn_mask_loss: 0.2399
Epoch 5/30
100/100 [==============================] - 148s 1s/step - loss: 0.4023 - rpn_class_loss: 0.0048 -
rpn_bbox_loss: 0.1381 - mrcnn_class_loss: 0.0190 - mrcnn_bbox_loss: 0.1208 - mrcnn_mask_loss:
0.1195 - val_loss: 0.9618 - val_rpn_class_loss: 0.0184 - val_rpn_bbox_loss: 0.3375 -
val_mrcnn_class_loss: 0.0387 - val_mrcnn_bbox_loss: 0.3339 - val_mrcnn_mask_loss: 0.2332
Epoch 6/30
100/100 [==============================] - 148s 1s/step - loss: 0.3663 - rpn_class_loss: 0.0039 -
rpn_bbox_loss: 0.1384 - mrcnn_class_loss: 0.0161 - mrcnn_bbox_loss: 0.1053 - mrcnn_mask_loss:
0.1027 - val_loss: 1.0122 - val_rpn_class_loss: 0.0172 - val_rpn_bbox_loss: 0.3712 -
val_mrcnn_class_loss: 0.0338 - val_mrcnn_bbox_loss: 0.3614 - val_mrcnn_mask_loss: 0.2286
Epoch 7/30
100/100 [==============================] - 148s 1s/step - loss: 0.3234 - rpn_class_loss: 0.0051 -
rpn_bbox_loss: 0.1344 - mrcnn_class_loss: 0.0164 - mrcnn_bbox_loss: 0.0665 - mrcnn_mask_loss:
0.1009 - val_loss: 1.1944 - val_rpn_class_loss: 0.0298 - val_rpn_bbox_loss: 0.5025 -
val_mrcnn_class_loss: 0.0317 - val_mrcnn_bbox_loss: 0.4013 - val_mrcnn_mask_loss: 0.2291
Epoch 8/30
100/100 [==============================] - 148s 1s/step - loss: 0.2871 - rpn_class_loss: 0.0038 -
rpn_bbox_loss: 0.1027 - mrcnn_class_loss: 0.0132 - mrcnn_bbox_loss: 0.0709 - mrcnn_mask_loss:
0.0966 - val_loss: 1.0217 - val_rpn_class_loss: 0.0246 - val_rpn_bbox_loss: 0.4587 -
val_mrcnn_class_loss: 0.0308 - val_mrcnn_bbox_loss: 0.2856 - val_mrcnn_mask_loss: 0.2220
Epoch 9/30
100/100 [==============================] - 148s 1s/step - loss: 0.2626 - rpn_class_loss: 0.0039 -
rpn_bbox_loss: 0.0892 - mrcnn_class_loss: 0.0117 - mrcnn_bbox_loss: 0.0675 - mrcnn_mask_loss:
0.0902 - val_loss: 0.9410 - val_rpn_class_loss: 0.0122 - val_rpn_bbox_loss: 0.3733 -
val_mrcnn_class_loss: 0.0378 - val_mrcnn_bbox_loss: 0.3065 - val_mrcnn_mask_loss: 0.2113
Epoch 10/30
100/100 [==============================] - 148s 1s/step - loss: 0.2396 - rpn_class_loss: 0.0036 -
rpn_bbox_loss: 0.0860 - mrcnn_class_loss: 0.0109 - mrcnn_bbox_loss: 0.0527 - mrcnn_mask_loss:
0.0864 - val_loss: 1.2221 - val_rpn_class_loss: 0.0275 - val_rpn_bbox_loss: 0.6394 -
val_mrcnn_class_loss: 0.0253 - val_mrcnn_bbox_loss: 0.2732 - val_mrcnn_mask_loss: 0.2567
Epoch 11/30
100/100 [==============================] - 148s 1s/step - loss: 0.2410 - rpn_class_loss: 0.0026 -
rpn_bbox_loss: 0.0870 - mrcnn_class_loss: 0.0118 - mrcnn_bbox_loss: 0.0562 - mrcnn_mask_loss:
0.0834 - val_loss: 1.0993 - val_rpn_class_loss: 0.0250 - val_rpn_bbox_loss: 0.5204 -
val_mrcnn_class_loss: 0.0236 - val_mrcnn_bbox_loss: 0.3260 - val_mrcnn_mask_loss: 0.2043
Epoch 12/30
100/100 [==============================] - 149s 1s/step - loss: 0.2275 - rpn_class_loss: 0.0026 -
rpn_bbox_loss: 0.0827 - mrcnn_class_loss: 0.0097 - mrcnn_bbox_loss: 0.0543 - mrcnn_mask_loss:
0.0782 - val_loss: 1.1595 - val_rpn_class_loss: 0.0169 - val_rpn_bbox_loss: 0.5786 -
val_mrcnn_class_loss: 0.0290 - val_mrcnn_bbox_loss: 0.3080 - val_mrcnn_mask_loss: 0.2271
Epoch 13/30
100/100 [==============================] - 148s 1s/step - loss: 0.2066 - rpn_class_loss: 0.0023 -
rpn_bbox_loss: 0.0694 - mrcnn_class_loss: 0.0152 - mrcnn_bbox_loss: 0.0421 - mrcnn_mask_loss:
0.0776 - val_loss: 1.4122 - val_rpn_class_loss: 0.0232 - val_rpn_bbox_loss: 0.8143 -
val_mrcnn_class_loss: 0.0440 - val_mrcnn_bbox_loss: 0.2980 - val_mrcnn_mask_loss: 0.2326
Epoch 14/30
```

```
100/100 [==============================] - 148s 1s/step - loss: 0.1942 - rpn_class_loss: 0.0030 -
rpn_bbox_loss: 0.0549 - mrcnn_class_loss: 0.0202 - mrcnn_bbox_loss: 0.0405 - mrcnn_mask_loss:
0.0757 - val_loss: 1.1397 - val_rpn_class_loss: 0.0232 - val_rpn_bbox_loss: 0.6276 -
val_mrcnn_class_loss: 0.0270 - val_mrcnn_bbox_loss: 0.2531 - val_mrcnn_mask_loss: 0.2088
Epoch 15/30
100/100 [==============================] - 148s 1s/step - loss: 0.1808 - rpn_class_loss: 0.0025 -
rpn_bbox_loss: 0.0524 - mrcnn_class_loss: 0.0176 - mrcnn_bbox_loss: 0.0364 - mrcnn_mask_loss:
0.0719 - val_loss: 1.2937 - val_rpn_class_loss: 0.0213 - val_rpn_bbox_loss: 0.6842 -
val_mrcnn_class_loss: 0.0409 - val_mrcnn_bbox_loss: 0.2964 - val_mrcnn_mask_loss: 0.2509
Epoch 16/30
100/100 [==============================] - 148s 1s/step - loss: 0.1678 - rpn_class_loss: 0.0022 -
rpn_bbox_loss: 0.0426 - mrcnn_class_loss: 0.0189 - mrcnn_bbox_loss: 0.0402 - mrcnn_mask_loss:
0.0640 - val_loss: 1.6118 - val_rpn_class_loss: 0.0399 - val_rpn_bbox_loss: 1.0040 -
val_mrcnn_class_loss: 0.0510 - val_mrcnn_bbox_loss: 0.3026 - val_mrcnn_mask_loss: 0.2142
Epoch 17/30
100/100 [==============================] - 148s 1s/step - loss: 0.1827 - rpn_class_loss: 0.0023 -
rpn_bbox_loss: 0.0488 - mrcnn_class_loss: 0.0189 - mrcnn_bbox_loss: 0.0423 - mrcnn_mask_loss:
0.0704 - val_loss: 0.8033 - val_rpn_class_loss: 0.0040 - val_rpn_bbox_loss: 0.2768 -
val_mrcnn_class_loss: 0.0620 - val_mrcnn_bbox_loss: 0.2366 - val_mrcnn_mask_loss: 0.2238
Epoch 18/30
100/100 [==============================] - 148s 1s/step - loss: 0.1615 - rpn_class_loss: 0.0025 -
rpn_bbox_loss: 0.0433 - mrcnn_class_loss: 0.0190 - mrcnn_bbox_loss: 0.0322 - mrcnn_mask_loss:
0.0645 - val_loss: 1.2665 - val_rpn_class_loss: 0.0226 - val_rpn_bbox_loss: 0.6697 -
val_mrcnn_class_loss: 0.0556 - val_mrcnn_bbox_loss: 0.2521 - val_mrcnn_mask_loss: 0.2664
Epoch 19/30
100/100 [==============================] - 148s 1s/step - loss: 0.1444 - rpn_class_loss: 0.0020 -
rpn_bbox_loss: 0.0412 - mrcnn_class_loss: 0.0171 - mrcnn_bbox_loss: 0.0217 - mrcnn_mask_loss:
0.0624 - val_loss: 1.3987 - val_rpn_class_loss: 0.0337 - val_rpn_bbox_loss: 0.8787 -
val_mrcnn_class_loss: 0.0545 - val_mrcnn_bbox_loss: 0.2214 - val_mrcnn_mask_loss: 0.2105
Epoch 20/30
100/100 [==============================] - 148s 1s/step - loss: 0.1544 - rpn_class_loss: 0.0031 -
rpn_bbox_loss: 0.0411 - mrcnn_class_loss: 0.0173 - mrcnn_bbox_loss: 0.0320 - mrcnn_mask_loss:
0.0609 - val_loss: 1.6057 - val_rpn_class_loss: 0.0411 - val_rpn_bbox_loss: 1.0134 -
val_mrcnn_class_loss: 0.0354 - val_mrcnn_bbox_loss: 0.2917 - val_mrcnn_mask_loss: 0.2241
Epoch 21/30
100/100 [==============================] - 148s 1s/step - loss: 0.1488 - rpn_class_loss: 0.0024 -
rpn_bbox_loss: 0.0431 - mrcnn_class_loss: 0.0163 - mrcnn_bbox_loss: 0.0269 - mrcnn_mask_loss:
0.0600 - val_loss: 0.9356 - val_rpn_class_loss: 0.0078 - val_rpn_bbox_loss: 0.3668 -
val_mrcnn_class_loss: 0.0609 - val_mrcnn_bbox_loss: 0.2423 - val_mrcnn_mask_loss: 0.2577
Epoch 22/30
100/100 [==============================] - 148s 1s/step - loss: 0.1455 - rpn_class_loss: 0.0019 -
rpn_bbox_loss: 0.0440 - mrcnn_class_loss: 0.0155 - mrcnn_bbox_loss: 0.0274 - mrcnn_mask_loss:
0.0567 - val_loss: 1.7164 - val_rpn_class_loss: 0.0383 - val_rpn_bbox_loss: 1.0650 -
val_mrcnn_class_loss: 0.0601 - val_mrcnn_bbox_loss: 0.2854 - val_mrcnn_mask_loss: 0.2676
Epoch 23/30
100/100 [==============================] - 149s 1s/step - loss: 0.1466 - rpn_class_loss: 0.0024 -
rpn_bbox_loss: 0.0475 - mrcnn_class_loss: 0.0157 - mrcnn_bbox_loss: 0.0244 - mrcnn_mask_loss:
0.0567 - val_loss: 1.7271 - val_rpn_class_loss: 0.0475 - val_rpn_bbox_loss: 1.0908 -
val_mrcnn_class_loss: 0.0392 - val_mrcnn_bbox_loss: 0.2951 - val_mrcnn_mask_loss: 0.2545
Epoch 24/30
100/100 [==============================] - 149s 1s/step - loss: 0.1415 - rpn_class_loss: 0.0015 -
rpn_bbox_loss: 0.0461 - mrcnn_class_loss: 0.0133 - mrcnn_bbox_loss: 0.0248 - mrcnn_mask_loss:
0.0558 - val_loss: 1.3441 - val_rpn_class_loss: 0.0227 - val_rpn_bbox_loss: 0.7374 -
val_mrcnn_class_loss: 0.0580 - val_mrcnn_bbox_loss: 0.2569 - val_mrcnn_mask_loss: 0.2691
Epoch 25/30
100/100 [==============================] - 149s 1s/step - loss: 0.1362 - rpn_class_loss: 0.0028 -
rpn_bbox_loss: 0.0394 - mrcnn_class_loss: 0.0099 - mrcnn_bbox_loss: 0.0251 - mrcnn_mask_loss:
0.0590 - val_loss: 1.1771 - val_rpn_class_loss: 0.0077 - val_rpn_bbox_loss: 0.4680 -
val_mrcnn_class_loss: 0.0590 - val_mrcnn_bbox_loss: 0.3987 - val_mrcnn_mask_loss: 0.2437
Epoch 26/30
100/100 [==============================] - 149s 1s/step - loss: 0.1211 - rpn_class_loss: 0.0020 -
rpn_bbox_loss: 0.0351 - mrcnn_class_loss: 0.0115 - mrcnn_bbox_loss: 0.0204 - mrcnn_mask_loss:
0.0521 - val_loss: 1.3103 - val_rpn_class_loss: 0.0234 - val_rpn_bbox_loss: 0.7374 -
val_mrcnn_class_loss: 0.0368 - val_mrcnn_bbox_loss: 0.2599 - val_mrcnn_mask_loss: 0.2528
Epoch 27/30
100/100 [==============================] - 149s 1s/step - loss: 0.1223 - rpn_class_loss: 0.0020 -
rpn_bbox_loss: 0.0334 - mrcnn_class_loss: 0.0121 - mrcnn_bbox_loss: 0.0199 - mrcnn_mask_loss:
0.0549 - val_loss: 1.9184 - val_rpn_class_loss: 0.0529 - val_rpn_bbox_loss: 1.2673 -
val_mrcnn_class_loss: 0.0516 - val_mrcnn_bbox_loss: 0.2897 - val_mrcnn_mask_loss: 0.2569
Epoch 28/30
```

```
100/100 [==============================] - 149s 1s/step - loss: 0.1121 - rpn_class_loss: 0.0017 -
rpn_bbox_loss: 0.0320 - mrcnn_class_loss: 0.0097 - mrcnn_bbox_loss: 0.0169 - mrcnn_mask_loss:
0.0518 - val_loss: 1.8875 - val_rpn_class_loss: 0.0371 - val_rpn_bbox_loss: 1.2870 -
val_mrcnn_class_loss: 0.0298 - val_mrcnn_bbox_loss: 0.2742 - val_mrcnn_mask_loss: 0.2594
Epoch 29/30
100/100 [==============================] - 148s 1s/step - loss: 0.1118 - rpn_class_loss: 0.0014 -
rpn_bbox_loss: 0.0312 - mrcnn_class_loss: 0.0101 - mrcnn_bbox_loss: 0.0178 - mrcnn_mask_loss:
0.0513 - val_loss: 1.4084 - val_rpn_class_loss: 0.0275 - val_rpn_bbox_loss: 0.7992 -
val_mrcnn_class_loss: 0.0357 - val_mrcnn_bbox_loss: 0.2687 - val_mrcnn_mask_loss: 0.2773
Epoch 30/30
100/100 [==============================] - 149s 1s/step - loss: 0.1198 - rpn_class_loss: 0.0018 -
rpn_bbox_loss: 0.0357 - mrcnn_class_loss: 0.0135 - mrcnn_bbox_loss: 0.0191 - mrcnn_mask_loss:
0.0498 - val_loss: 1.7065 - val_rpn_class_loss: 0.0347 - val_rpn_bbox_loss: 1.0505 -
val_mrcnn_class_loss: 0.0509 - val_mrcnn_bbox_loss: 0.2679 - val_mrcnn_mask_loss: 0.3024
```

Result?

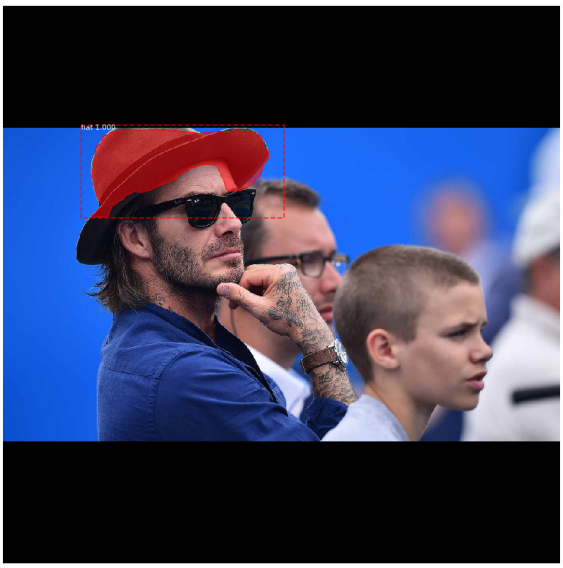
I choose to use epoch29.h5.

Train Dataset	Val Dataset
	
	

['BG', 'hat']

['BG', 'hat']

['BG', 'hat']



The image result is good.

TEST

Test subject: 5/5 recognized.(using splash)

Video Track

URL:https://drive.google.com/open?id=1XlwOitDnXFE0SpzzVFg_SUidUHKwDKDP

It is no so good. But sometime is very impressive!

I set confidence >0.95 will display a box but if it is “might” a hat it will splash that.

Code and data structure

first, it is on my GitHub.
https://github.com/Waxpple/coco_model
And there is my data structure and some co lab file.
After training finished, the file is too large to download, so I upload it on google drive.
https://drive.google.com/file/d/1L-xyHzHNsQhPGFXdi2WbjZ2_lE4BPmzF/view?usp=sharing

If want to use hat.py to splash img online.
Use python3 hat2.py splash —weight= “path to epoch29.h5” —image or —video=url.
*video can use visualize instead of splash. Visualize include anchor.

The weights: https://drive.google.com/drive/folders/1VreAMyPV9VDYY9G-S7PrkQKBHgmMhyUQ?usp=sharing

Reference

COCO - Common Objects in Context
http://cocodataset.org/#explore

Object Detection using TensorFlow and COCO Pre-Trained Models
https://medium.com/object-detection-using-tensorflow-and-coco-pre/object-detection-using-tensorflow-and-coco-pre-trained-models-5d8386019a8

VGG Image Annotator
https://www.robots.ox.ac.uk/~vgg/software/via/via_demo.html

via_export_coco.json
file:///Users/ethan/Downloads/via_export_coco.json

Online JSON Viewer
http://jsonviewer.stack.hu/

How to create custom COCO data set for instance segmentation | DLology
https://www.dlology.com/blog/how-to-create-custom-coco-data-set-for-instance-segmentation/

Online JSON Viewer
http://jsonviewer.stack.hu/

Find Images in Online Databases - Image Resources - Research Guides at UCLA Library
https://guides.library.ucla.edu/c.php?g=180361&p=1185038

How to (quickly) build a deep learning image dataset - PyImageSearch
https://www.pyimagesearch.com/2018/04/09/how-to-quickly-build-a-deep-learning-image-dataset/

Bing Image Search API | Microsoft Azure
https://azure.microsoft.com/en-us/services/cognitive-services/bing-image-search-api/

Waxpple/coco_model
https://github.com/Waxpple/coco_model

How to Use Google Colaboratory for Video Processing - DZone AI
https://dzone.com/articles/how-to-use-google-colaboratory-for-video-processin

Splash of Color: Instance Segmentation with Mask R-CNN and TensorFlow
https://engineering.matterport.com/splash-of-color-instance-segmentation-with-mask-r-cnn-and-tensorflow-7c761e238b46

Mask_RCNN/samples/balloon at master · matterport/Mask_RCNN
https://github.com/matterport/Mask_RCNN/tree/master/samples/balloon

Apply splash effect on a video. Requires OpenCV 3.2 nrcnn - Google 搜尋
https://www.google.com/search?sxsrf=ACYBGNQr3Mca9y49804JQsxbbhHE-WXR0w%3A1573742340713&ei=BGfNXbiZK5L4U67RvPgB&q=Apply+splash+effect+on+a+video.+Requires+OpenCV+3.2+nrcnn&oq=Apply+splash+effect+on+a+video.+Requires+OpenCV+3.2+nrcnn&gs_l=psy-ab.3..33i160l2.4267.5504..5691...0.0..0.103.534.5j1......0....1..gws-wiz.......35i302i39.DKkkPW-31YQ&ved=0ahUKEwi4q76J9-nlAhUS_BQKHa4oDx8Q4dUDCAs&uact=5

Untitled11.ipynb - Colaboratory
https://colab.research.google.com/drive/1Li8pKJ5PacgSL9nn6gwxfFFh0CRYIQuX#scrollTo=mCnfwa2Dms8a

python instead of call command line - Google 搜尋
https://www.google.com/search?q=python+instead+of+call+command+line&oq=python+instead+of+call+command+line&aqs=chrome..69i57.7677j0j4&sourceid=chrome&ie=UTF-8

deep learning - Import Matterport's Mask-RCNN model from github - error:ZipImportError: bad local file header - Stack Overflow
https://stackoverflow.com/questions/55378668/import-matterports-mask-rcnn-model-from-github-errorzipimporterror-bad-loca

python - Loading images in google colab - Stack Overflow
https://stackoverflow.com/questions/49186905/loading-images-in-google-colab

Balloon Sample: What if we want to train for more than one class · Issue #372 · matterport/Mask_RCNN
https://github.com/matterport/Mask_RCNN/issues/372

Mask R-CNN 训练自己的数据集（balloon过程+报错解释） - 刘大鱼的博客 - CSDN博客
https://blog.csdn.net/qq_40013071/article/details/88384567

Upload files · Waxpple/coco_model
https://github.com/Waxpple/coco_model/upload/master

Untitled13.ipynb - Colaboratory
https://colab.research.google.com/drive/1EIK_hfIFUxecQZoDewbGiAfNCnEH77PC

isepiot | etherpad.net Pad
https://etherpad.net/p/isepiot

How to Train an Object Detection Model with Keras
https://machinelearningmastery.com/how-to-train-an-object-detection-model-with-keras/

python 3.x - How to download large files (like weights of a model) from Colaboratory? - Stack Overflow
https://stackoverflow.com/questions/49428332/how-to-download-large-files-like-weights-of-a-model-from-colaboratory

python-3.x: 如何从Google Colab下载多个文件或整个文件夹？
https://xbuba.com/questions/50453428

hat crowd - Google 搜尋
https://www.google.com/search?q=hat+crowd&tbm=isch&sxsrf=ACYBGNRVtJ14s2HkEhBFIh00Xbe8mG7T0Q:1573742746189&source=lnt&tbs=isz:l&sa=X&ved=0ahUKEwjM0urK-OnlAhVcXRUIHXL2CsgQpwUIJA&biw=1645&bih=791&dpr=2#imgrc=4LEYfZOjIqlIxM:

Success code=4/tAHDN9R0owTQNAFQVvZrUrcX964HCLaqcLGGMTV7BYZzmixdU3M0sOM&scope=email%20https://www.googleapis.com/auth/userinfo.email%20https://www.googleapis.com/auth/cloud-platform%20https://www.googleapis.com/auth/appengine.admin%20https://www.googleapis.com/auth/compute%20https://www.googleapis.com/auth/accounts.reauth%20https://www.googleapis.com/auth/drive%20openid&authuser=1&hd=gapps.ntust.edu.tw&session_state=dce059445ff6f1cc7206ca14db02724cde1d9a75..d5e9&prompt=consent

COCO - Common Objects in Context
http://cocodataset.org/#explore

Object Detection using TensorFlow and COCO Pre-Trained Models
https://medium.com/object-detection-using-tensorflow-and-coco-pre/object-detection-using-tensorflow-and-coco-pre-trained-models-5d8386019a8

VGG Image Annotator
https://www.robots.ox.ac.uk/~vgg/software/via/via_demo.html

via_export_coco.json
file:///Users/ethan/Downloads/via_export_coco.json

Online JSON Viewer
http://jsonviewer.stack.hu/

How to create custom COCO data set for instance segmentation | DLology
https://www.dlology.com/blog/how-to-create-custom-coco-data-set-for-instance-segmentation/

Online JSON Viewer
http://jsonviewer.stack.hu/

Find Images in Online Databases - Image Resources - Research Guides at UCLA Library
https://guides.library.ucla.edu/c.php?g=180361&p=1185038

How to (quickly) build a deep learning image dataset - PyImageSearch
https://www.pyimagesearch.com/2018/04/09/how-to-quickly-build-a-deep-learning-image-dataset/

Bing Image Search API | Microsoft Azure
https://azure.microsoft.com/en-us/services/cognitive-services/bing-image-search-api/

Waxpple/coco_model: abc/cocolab
https://github.com/Waxpple/coco_model

How to Use Google Colaboratory for Video Processing - DZone AI
https://dzone.com/articles/how-to-use-google-colaboratory-for-video-processin

Splash of Color: Instance Segmentation with Mask R-CNN and TensorFlow
https://engineering.matterport.com/splash-of-color-instance-segmentation-with-mask-r-cnn-and-tensorflow-7c761e238b46

Mask_RCNN/samples/balloon at master · matterport/Mask_RCNN
https://github.com/matterport/Mask_RCNN/tree/master/samples/balloon

Untitled11.ipynb - Colaboratory
https://colab.research.google.com/drive/1Li8pKJ5PacgSL9nn6gwxfFFh0CRYIQuX#scrollTo=mCnfwa2Dms8a

python instead of call command line - Google 搜尋
https://www.google.com/search?q=python+instead+of+call+command+line&oq=python+instead+of+call+command+line&aqs=chrome..69i57.7677j0j4&sourceid=chrome&ie=UTF-8

deep learning - Import Matterport's Mask-RCNN model from github - error:ZipImportError: bad local file header - Stack Overflow
https://stackoverflow.com/questions/55378668/import-matterports-mask-rcnn-model-from-github-errorzipimporterror-bad-loca

python - Loading images in google colab - Stack Overflow
https://stackoverflow.com/questions/49186905/loading-images-in-google-colab

Balloon Sample: What if we want to train for more than one class · Issue #372 · matterport/Mask_RCNN
https://github.com/matterport/Mask_RCNN/issues/372

Mask R-CNN 训练自己的数据集（balloon过程+报错解释） - 刘大鱼的博客 - CSDN博客
https://blog.csdn.net/qq_40013071/article/details/88384567

Waxpple/coco_model: abc/cocolab
https://github.com/Waxpple/coco_model