In [10]:
```
# this is the YAML file Roboflow wrote for us that we're loading into this notebook with our data
%cat data.yaml
```

```
train: ../train/images
val: ../valid/images

nc: 7
names: ['multi','rain','fragile','hang','up','handle','cut']
```

Inspect Model Configuration and Architecture

Let's look at the Scaled-YOLOv4 Configuration architecture

In [11]:
```
%cat /content/ScaledYOLOv4/models/yolov4-csp.yaml
```

```
# parameters
nc: 80  # number of classes
depth_multiple: 1.0  # model depth multiple
width_multiple: 1.0  # layer channel multiple

# anchors
anchors:
  - [12,16, 19,36, 40,28]  # P3/8
  - [36,75, 76,55, 72,146]  # P4/16
  - [142,110, 192,243, 459,401]  # P5/32

# yolov4-csp backbone
backbone:
  # [from, number, module, args]
  [[-1, 1, Conv, [32, 3, 1]],  # 0
   [-1, 1, Conv, [64, 3, 2]],  # 1-P1/2
   [-1, 1, Bottleneck, [64]],
   [-1, 1, Conv, [128, 3, 2]],  # 3-P2/4
   [-1, 2, BottleneckCSP, [128]],
   [-1, 1, Conv, [256, 3, 2]],  # 5-P3/8
   [-1, 8, BottleneckCSP, [256]],
   [-1, 1, Conv, [512, 3, 2]],  # 7-P4/16
   [-1, 8, BottleneckCSP, [512]],
   [-1, 1, Conv, [1024, 3, 2]], # 9-P5/32
   [-1, 4, BottleneckCSP, [1024]],  # 10
  ]

# yolov4-csp head
# na = len(anchors[0])
head:
  [[-1, 1, SPPCSP, [512]], # 11
   [-1, 1, Conv, [256, 1, 1]],
   [-1, 1, nn.Upsample, [None, 2, 'nearest']],
   [8, 1, Conv, [256, 1, 1]], # route backbone P4
   [[-1, -2], 1, Concat, [1]],
   [-1, 2, BottleneckCSP2, [256]], # 16
   [-1, 1, Conv, [128, 1, 1]],
   [-1, 1, nn.Upsample, [None, 2, 'nearest']],
   [6, 1, Conv, [128, 1, 1]], # route backbone P3
   [[-1, -2], 1, Concat, [1]],
   [-1, 2, BottleneckCSP2, [128]], # 21
   [-1, 1, Conv, [256, 3, 1]],
```

```
   [-2, 1, Conv, [256, 3, 2]],
   [[-1, 16], 1, Concat, [1]],  # cat
   [-1, 2, BottleneckCSP2, [256]], # 25
   [-1, 1, Conv, [512, 3, 1]],
   [-2, 1, Conv, [512, 3, 2]],
   [[-1, 11], 1, Concat, [1]],  # cat
   [-1, 2, BottleneckCSP2, [512]], # 29
   [-1, 1, Conv, [1024, 3, 1]],

   [[22,26,30], 1, Detect, [nc, anchors]],   # Detect(P3, P4, P5)
  ]
```

Train Custom Scaled-YOLOv4 Detector

Next, we'll fire off training!

Here, we are able to pass a number of arguments:

- **img:** define input image size
- **batch:** determine batch size
- **epochs:** define the number of training epochs. (Note: often, 3000+ are common here!)
- **data:** set the path to our yaml file
- **cfg:** specify our model configuration
- **weights:** specify a custom path to weights.
- **name:** result names
- **nosave:** only save the final checkpoint
- **cache:** cache images for faster training

In [12]:
```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

In [14]:
```
# train scaled-YOLOv4 on custom data for 100 epochs
# time its performance
%%time
%cd /content/ScaledYOLOv4/
!python train.py --img 416 --batch 16 --epochs 100 --data '../data.yaml'
--cfg ./models/yolov4-csp.yaml --weights '/content/drive/MyDrive/yolo_wei
ghts/yolov4-p5_.pt' --name yolov4-csp-results  --cache
```

```
/content/ScaledYOLOv4
Using CUDA device0 _CudaDeviceProperties(name='Tesla T4', total_memory=15
109MB)

Namespace(adam=False, batch_size=16, bucket='', cache_images=True, cfg
='./models/yolov4-csp.yaml', data='../data.yaml', device='', epochs=100,
evolve=False, global_rank=-1, hyp='data/hyp.finetune.yaml', img_size=[41
6, 416], local_rank=-1, logdir='runs/', multi_scale=False, name='yolov4-c
sp-results', noautoanchor=False, nosave=False, notest=False, rect=False,
resume=False, single_cls=False, sync_bn=False, total_batch_size=16, weigh
ts='/content/drive/MyDrive/yolo_weights/yolov4-p5_.pt', world_size=1)
Start Tensorboard with "tensorboard --logdir runs/", view at http://local
host:6006/
Hyperparameters {'lr0': 0.01, 'momentum': 0.937, 'weight_decay': 0.0005,
'giou': 0.05, 'cls': 0.5, 'cls_pw': 1.0, 'obj': 1.0, 'obj_pw': 1.0, 'iou_
t': 0.2, 'anchor_t': 4.0, 'fl_gamma': 0.0, 'hsv_h': 0.015, 'hsv_s': 0.7,
'hsv_v': 0.4, 'degrees': 0.0, 'translate': 0.5, 'scale': 0.8, 'shear': 0.
```

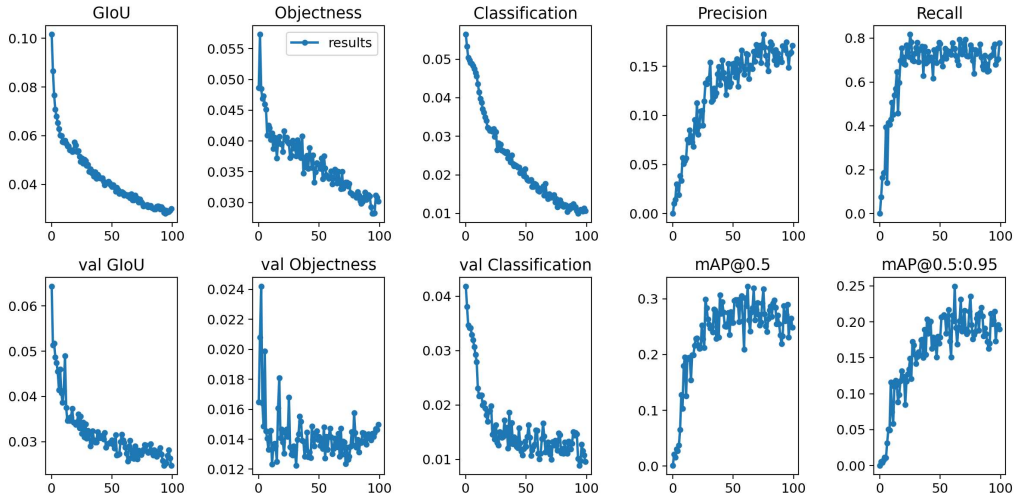
```
0, 'perspective': 0.0, 'flipud': 0.0, 'fliplr': 0.5, 'mixup': 0.2}
Overriding ./models/yolov4-csp.yaml nc=80 with nc=7

                from  n    params  module
arguments
  0                -1  1       928  models.common.Conv
[3, 32, 3, 1]
  1                -1  1     18560  models.common.Conv
[32, 64, 3, 2]
  2                -1  1     20672  models.common.Bottleneck
[64, 64]
  3                -1  1     73984  models.common.Conv
[64, 128, 3, 2]
  4                -1  1    119936  models.common.BottleneckCSP
[128, 128, 2]
  5                -1  1    295424  models.common.Conv
[128, 256, 3, 2]
  6                -1  1   1463552  models.common.BottleneckCSP
[256, 256, 8]
  7                -1  1   1180672  models.common.Conv
[256, 512, 3, 2]
  8                -1  1   5843456  models.common.BottleneckCSP
[512, 512, 8]
  9                -1  1   4720640  models.common.Conv
[512, 1024, 3, 2]
 10                -1  1  12858368  models.common.BottleneckCSP
[1024, 1024, 4]
 11                -1  1   7610368  models.common.SPPCSP
[1024, 512, 1]
 12                -1  1    131584  models.common.Conv
[512, 256, 1, 1]
 13                -1  1         0  torch.nn.modules.upsampling.Upsample
[None, 2, 'nearest']
 14                 8  1    131584  models.common.Conv
[512, 256, 1, 1]
 15          [-1, -2]  1         0  models.common.Concat
[1]
 16                -1  1   1642496  models.common.BottleneckCSP2
[512, 256, 2]
 17                -1  1     33024  models.common.Conv
[256, 128, 1, 1]
 18                -1  1         0  torch.nn.modules.upsampling.Upsample
[None, 2, 'nearest']
 19                 6  1     33024  models.common.Conv
[256, 128, 1, 1]
 20          [-1, -2]  1         0  models.common.Concat
[1]
 21                -1  1    411648  models.common.BottleneckCSP2
[256, 128, 2]
 22                -1  1    295424  models.common.Conv
[128, 256, 3, 1]
 23                -2  1    295424  models.common.Conv
[128, 256, 3, 2]
 24          [-1, 16]  1         0  models.common.Concat
[1]
 25                -1  1   1642496  models.common.BottleneckCSP2
[512, 256, 2]
 26                -1  1   1180672  models.common.Conv
[256, 512, 3, 1]
 27                -2  1   1180672  models.common.Conv
[256, 512, 3, 2]
 28          [-1, 11]  1         0  models.common.Concat
```

Note from Glenn: Partially completed `results.txt` files can be plotted with `from utils.utils import plot_results; plot_results()`.

In [15]:
```
# Start tensorboard
# Launch after you have started training
# logs save in the folder "runs"
%load_ext tensorboard
%tensorboard --logdir runs
```

```
<IPython.core.display.Javascript object>
```

In [18]:
```
# we can also output some older school graphs if the tensor board isn't w
orking for whatever reason...
#from utils.general import plot_results  # plot results.txt as results.pn
g
from IPython.display import Image, display
display(Image('/content/ScaledYOLOv4/runs/exp1_yolov4-csp-results/result
s.png'))  # view results.png
```



Curious? Visualize Our Training Data with Labels

After training starts, view `train*.jpg` images to see training images, labels and augmentation effects.

Note a mosaic dataloader is used for training (shown below), a new dataloading concept developed by Glenn Jocher and first featured in YOLOv4 (https://arxiv.org/abs/2004.10934).

In [21]:
```
# first, display our ground truth data
print("GROUND TRUTH TRAINING DATA:")
Image(filename='/content/ScaledYOLOv4/runs/exp1_yolov4-csp-results/test_b
atch0_gt.jpg', width=900)
```

```
GROUND TRUTH TRAINING DATA:
```

Out[21]:

In [22]:
```
# print out an augmented training example
print("GROUND TRUTH AUGMENTED TRAINING DATA:")
Image(filename='/content/ScaledYOLOv4/runs/exp1_yolov4-csp-results/train_batch0.jpg', width=900)
```

```
GROUND TRUTH AUGMENTED TRAINING DATA:
```

Out[22]:

Run Inference With Trained Weights

Run inference with a pretrained checkpoint on contents of `test/images` folder downloaded from Roboflow.

In [23]:
```
# trained weights are saved by default in our weights folder
%ls runs/
```

```
exp0_yolov4-csp-results/  exp1_yolov4-csp-results/
```

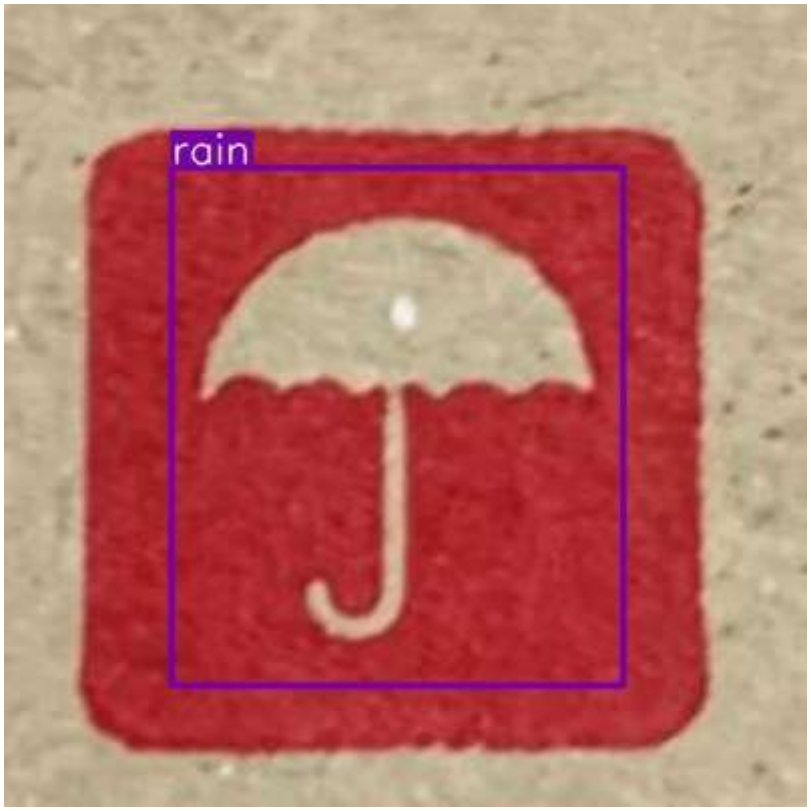
```
image 70/73 /content/resized_test/resized_test71.jpg: 416x416 1 rains, Do
ne. (0.015s)
image 71/73 /content/resized_test/resized_test72.jpg: 416x416 Done. (0.01
5s)
image 72/73 /content/resized_test/resized_test8.jpg: 416x416 Done. (0.015
s)
image 73/73 /content/resized_test/resized_test9.jpg: 416x416 Done. (0.013
s)
Results saved to inference/output
Done. (1.596s)
```

In [40]:
```
#display inference on ALL test images
#this looks much better with longer training above
import glob
from IPython.display import Image, display

for imageName in glob.glob('./inference/output/*.jpg'): #assuming JPG
    display(Image(filename=imageName))
    print("₩n")
```

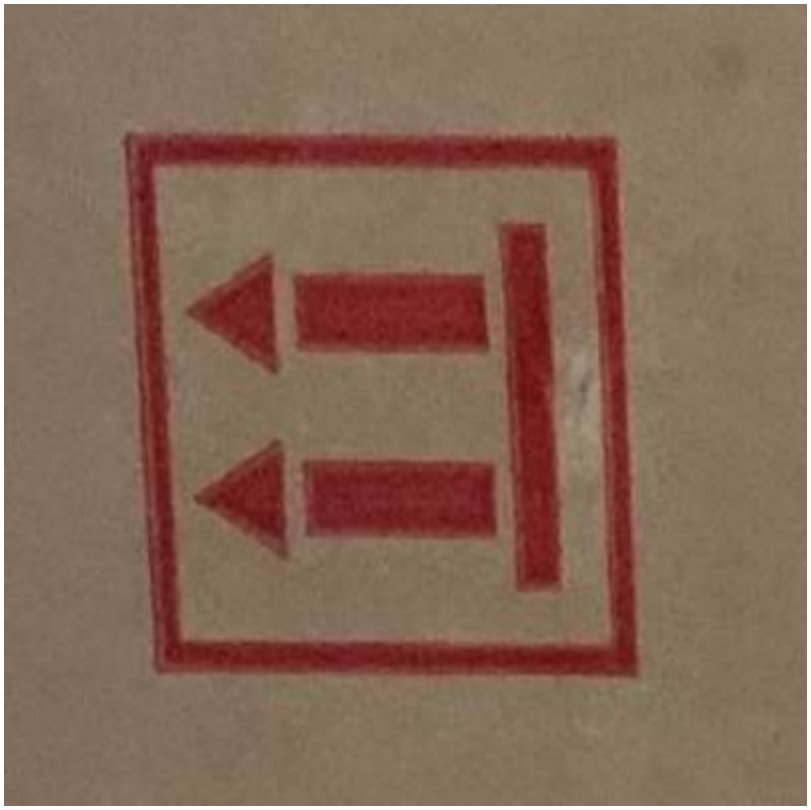

up


handle


습기조심


rain


파손주의
절대 던지지 마세요!
HANDLE WITH CARE




up


식품긴급배송
긴급배송 바랍니다!
rain
당일배송
www.dutch-coffee.co.kr


습기조심
칼사용조심


up
rain
fragile
AP-1818C
AP-1818C


rain
피그미 매운맛
(주)에이스엠앤티
TEL : 02-2283-1131


fragile
깨짐주의


깨짐주의
개봉시 칼을 사용하지 마세요
제품 손상의 위험이 있습니다


hang
rain
up
handle
1577-7011
coupang
208 B02