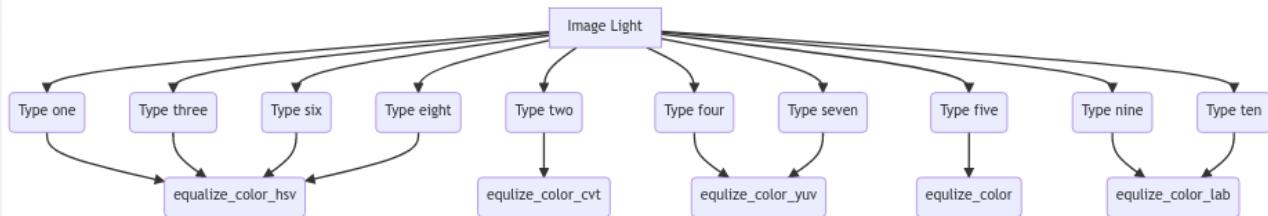


Histogram equalization

is the way we choose to equalize the intensity of the images, [OpenCV](#) has Implementation of it which handles gray images, so according to this [StackoverFlow post](#) in RGB images each Chanel represents the **intensity of the related color**, so we saturated the **Brightness** of the image, By using **standardized** colors paces that encode brightness and color **separately** like (YCbCr, HSV, ect).



and then **Apply simple Histogram**, because Histogram lacking results, due to its lack of rag rad to outliers and the location of the pixel, we chose to implement Contrast Limited Adaptive HE witch have better slightly better RESTPlus.

results







techniques used in each task

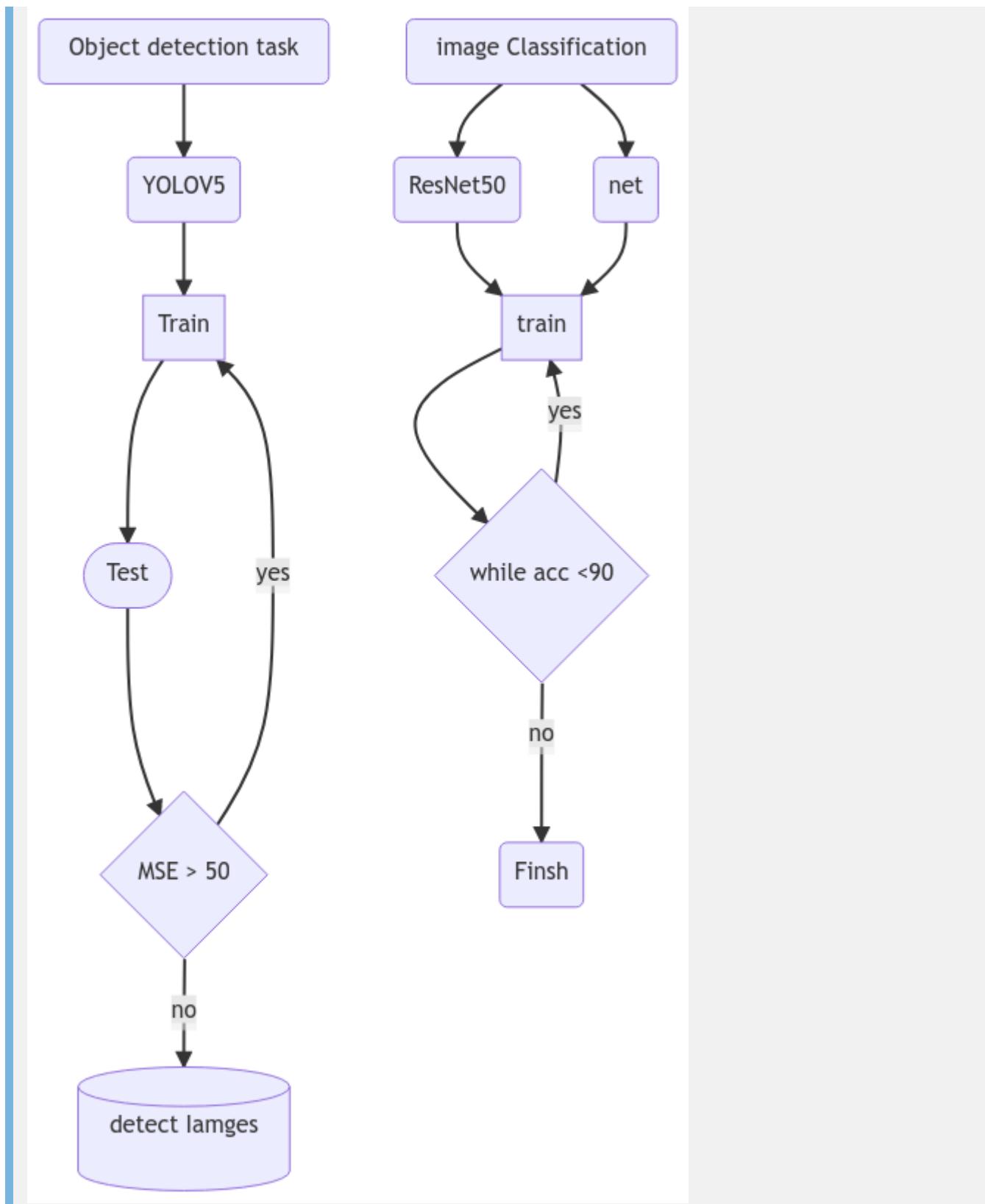
task 1 --> Contrast Limited Adaptive HE

task 2 --> (resnet50 - imageNet)

task 3 --> yolov5

task 4 --> k-mean, mean shift

Training and Testing times.



YOLOV5 --> 18

ImageNET 5

ResNet50 4

k-means and mean shift --> 5

- Image Net --> 91 acc
- YOLO5

```
running...  
Model summary: 213 layers, 7042489 parameters, 0 gradients, 15.9 GFLOPS  
    Class   Images   Labels      P      R  mAP@.5: .95: 100% 94/94 [00:31<00:00,  3.01it/s]  
    all     3000    10381    0.994    0.993    0.995    0.843  
    Bicycle 3000     463    0.996      1    0.995    0.856  
    Boat    3000     530     0.99    0.998    0.995    0.821  
    Bottle   3000     654    0.987    0.998    0.995    0.817  
    Bus     3000     333    0.995      1    0.995    0.894  
    Car     3000    1238    0.993    0.994    0.995    0.841  
    Cat     3000     311    0.997    0.994    0.994    0.866  
    Chair   3000    1138    0.994    0.983    0.995    0.853  
    Cup     3000     864    0.996    0.999    0.995    0.827  
    Dog     3000     335    0.996      1    0.995    0.871  
    Motorbike 3000     490    0.998    0.989    0.995    0.84  
    People   3000    3271    0.994    0.989    0.995    0.806  
    Table   3000     754    0.993    0.969    0.992    0.823  
Results saved to runs/train/exp18
```

wandb: Waiting for W&B process to finish... (success).

wandb: Run history:

wandb: metrics/mAP_0.5

wandb: metrics/mAP_0.5:0.95

wandb: metrics/precision

wandb: metrics/recall

wandb: train/box_loss

wandb: train/cls_loss

wandb: train/obj_loss

wandb: val/box_loss

wandb: val/cls_loss

wandb: val/obj_loss

wandb: x/lr0

wandb: x/lr1

wandb: x/lr2

wandb:
..

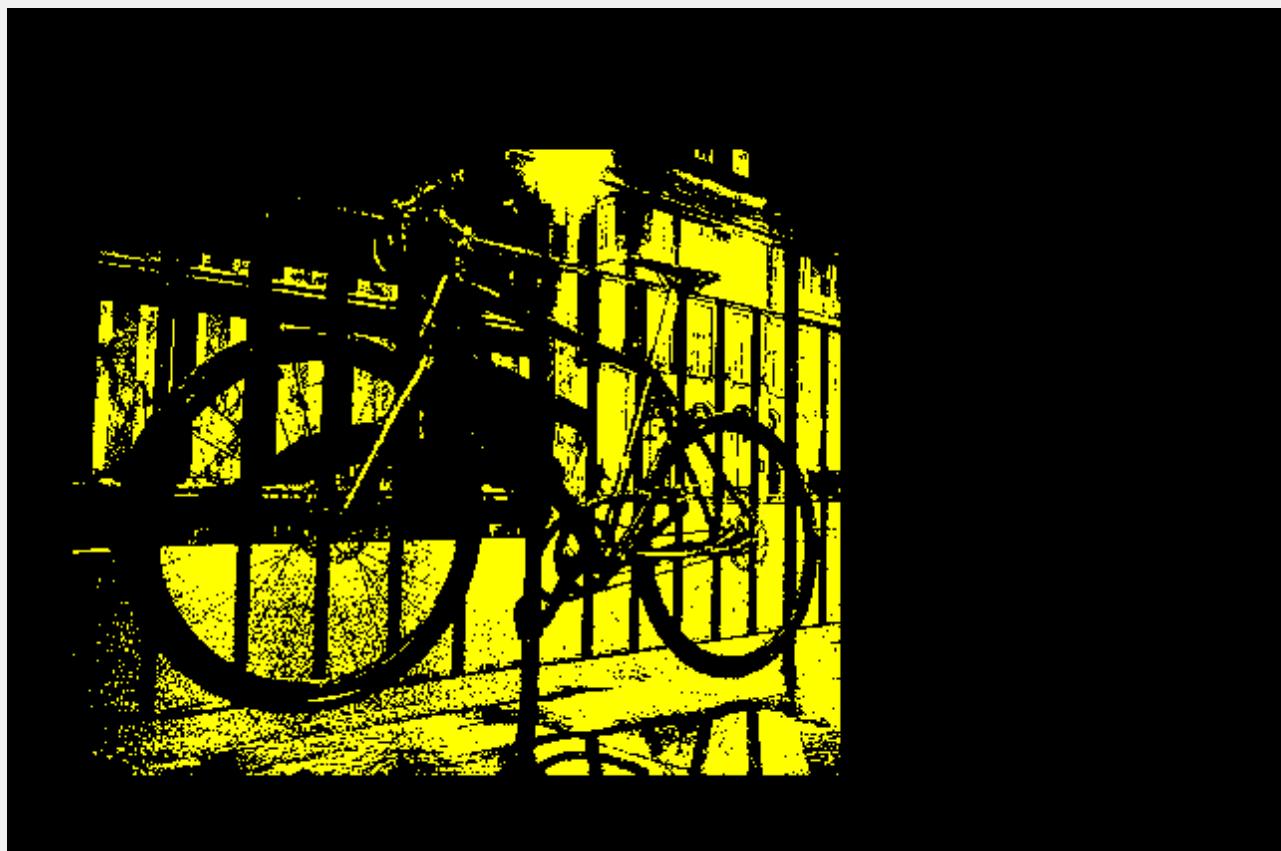


```
wandb: Run summary:  
wandb:     best/epoch 37  
wandb:     best/mAP_0.5 0.99458  
wandb:     best/mAP_0.5:0.95 0.84265  
wandb:     best/precision 0.99408  
wandb:     best/recall 0.99286  
wandb:     metrics/mAP_0.5 0.99458  
wandb:     metrics/mAP_0.5:0.95 0.84274  
wandb:     metrics/precision 0.99413  
wandb:     metrics/recall 0.99284  
wandb:     train/box_loss 0.0232  
wandb:     train/cls_loss 0.00286  
wandb:     train/obj_loss 0.02283  
wandb:     val/box_loss 0.01784  
wandb:     val/cls_loss 0.00085  
wandb:     val/obj_loss 0.01202  
wandb:             x/lr0 0.0006  
wandb:             x/lr1 0.0006  
wandb:             x/lr2 0.0006  
wandb:
```

Overview	Data	Code	Discussion	Leaderboard	Rules	Team	My Submissions	Late Submission	...
6	T2						0.42578	18	1d
7	T6						0.43192	63	3d
8	T1						0.44125	7	2d
9	sara mahmoud						0.46680	6	3d
10	T24						0.49459	9	5d
11	T10						0.50068	2	10h
12	Nada Hany						0.50676	3	5d
13	T_19						0.50831	32	7d

- segmentation

Results







- Object detection

