Hard-Hat Dataset

Dataset :

Link: https://public.roboflow.com/object-detection/hard-hat-workers

Data Distribution :

Train dataset : 100 Images
Validation dataset : 500 Images

Classes :

1. Hat
2. No-Hat

Annotation Format: MS-COCO

Model :

Name: Cascade Mask-RCNN with Swin-Tiny backbone and 1x GPU support

Training :

Environment: Google Colab with Tesla-V4 GPU (Linux)

Scheduler :

Epochs : 36 (3x of Mask-RCNN runtime, best possible baseline
Samples_per_gpu : 2
Workers_per_gpu : 2

Method :

In this experiment, the few-shot learning is adopted.
So, he training set has been opted as the validation /test and vice versa.

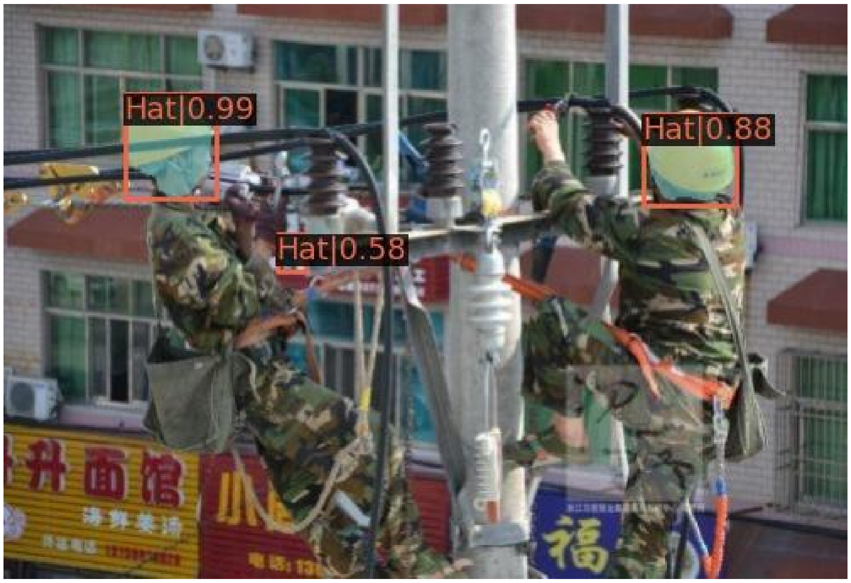
Colab Link :

https://colab.research.google.com/drive/1jtD16Gf5ciXh2-dS-Ih314Oi3Fp2nUn4?usp=sharing

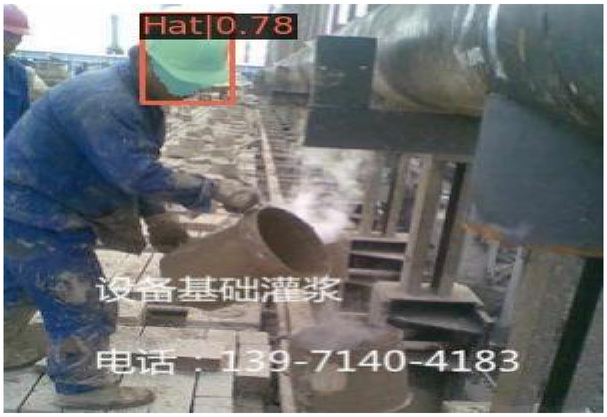
Evaluation :

```
Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.199
Average Precision  (AP) @[ IoU=0.50      | area=   all | maxDets=1000 ] = 0.483
Average Precision  (AP) @[ IoU=0.75      | area=   all | maxDets=1000 ] = 0.126
Average Precision  (AP) @[ IoU=0.50:0.95 | area= small | maxDets=1000 ] = 0.193
Average Precision  (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=1000 ] = 0.231
Average Precision  (AP) @[ IoU=0.50:0.95 | area= large | maxDets=1000 ] = 0.090
Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.334
Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=300 ] = 0.334
Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=1000 ] = 0.334
Average Recall     (AR) @[ IoU=0.50:0.95 | area= small | maxDets=1000 ] = 0.307
Average Recall     (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=1000 ] = 0.388
Average Recall     (AR) @[ IoU=0.50:0.95 | area= large | maxDets=1000 ] = 0.165

Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.184
Average Precision  (AP) @[ IoU=0.50      | area=   all | maxDets=1000 ] = 0.477
Average Precision  (AP) @[ IoU=0.75      | area=   all | maxDets=1000 ] = 0.103
Average Precision  (AP) @[ IoU=0.50:0.95 | area= small | maxDets=1000 ] = 0.177
Average Precision  (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=1000 ] = 0.215
Average Precision  (AP) @[ IoU=0.50:0.95 | area= large | maxDets=1000 ] = 0.092
Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.305
Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=300 ] = 0.305
Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=1000 ] = 0.305
Average Recall     (AR) @[ IoU=0.50:0.95 | area= small | maxDets=1000 ] = 0.277
Average Recall     (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=1000 ] = 0.358
Average Recall     (AR) @[ IoU=0.50:0.95 | area= large | maxDets=1000 ] = 0.169
```

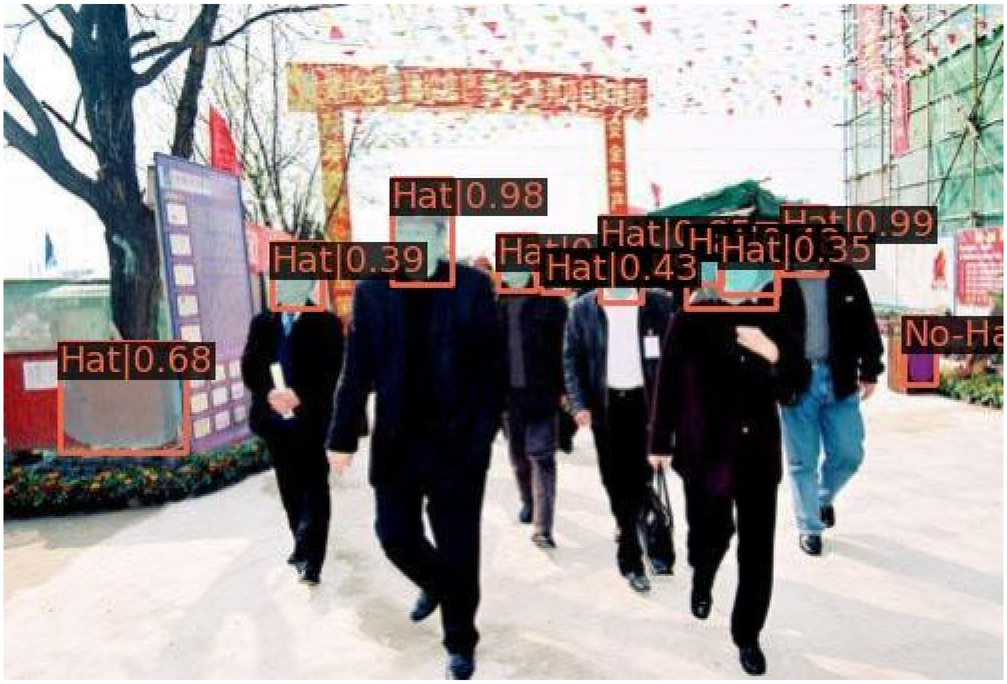
Test Results and explanation :



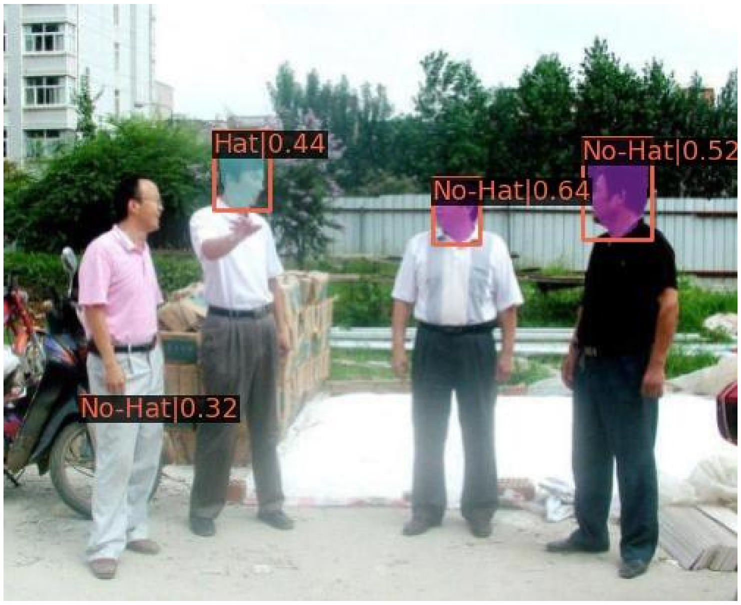
We can see that hats are correctly identified in this image.



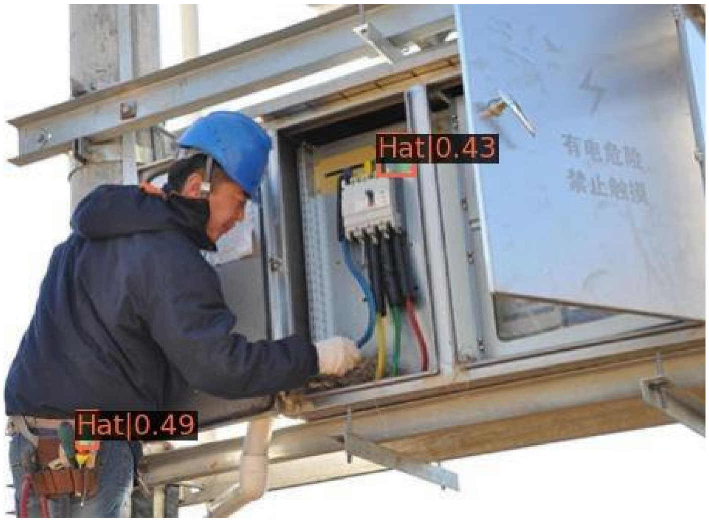
This image predicts correct bounding boxes and class , still the face of the worker has not been correctly segmented. This can be explained as the abruption of pixel value gradient, the color of the hardhat and the face are quite different thus, the attention layer leaves the pixels from the face.



In this picture we can see that the hardhat wearing people are correctly identified, still, some errors are present with the neighbouring different objects which look like helmets. We can see the object on the left side of the image predicted as “Hat”.



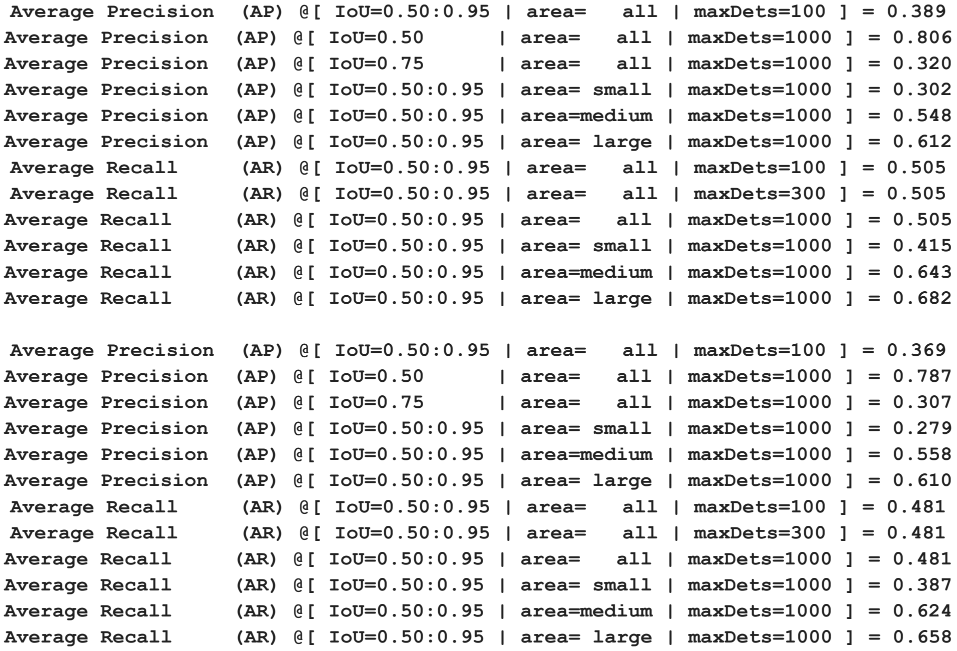
But the actual problem, with no-hat class, is that it tends to objectify it as the hat class, thus lowering the accuracy. These phenomena we can see there are some objects which are also classified and bounded by the boxes . Lowering the optimizer learning rate increases this phenomena. The picture shown below is a perfect example of that.



Improvised models like DCN were chosen as the next. DCN's deformable nature was to extract the spatially deformed characters of the objects.

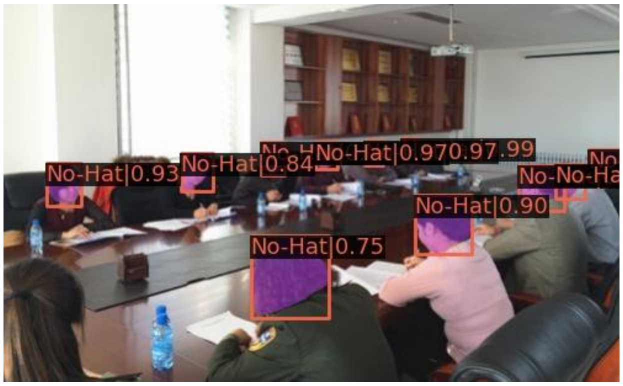
Scheduler: x1 (12 epochs)

Results :

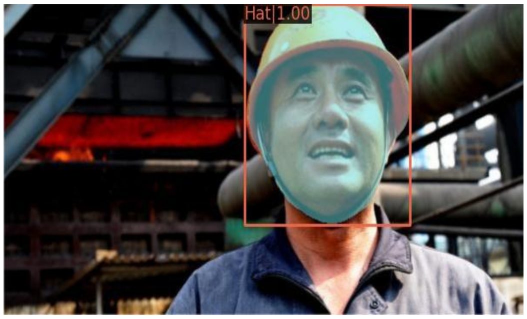


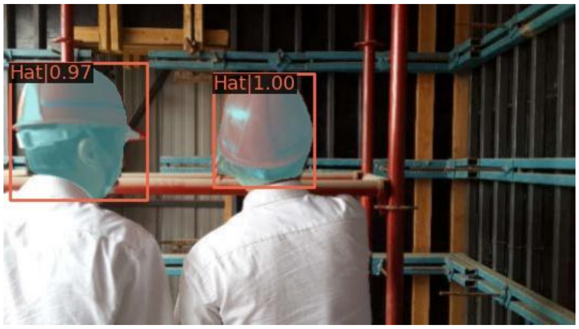
```
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.389
Average Precision  (AP) @[ IoU=0.50      | area=   all | maxDets=1000 ] = 0.806
Average Precision  (AP) @[ IoU=0.75      | area=   all | maxDets=1000 ] = 0.320
Average Precision  (AP) @[ IoU=0.50:0.95 | area= small | maxDets=1000 ] = 0.302
Average Precision  (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=1000 ] = 0.548
Average Precision  (AP) @[ IoU=0.50:0.95 | area= large | maxDets=1000 ] = 0.612
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.505
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=300 ] = 0.505
Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=1000 ] = 0.505
Average Recall     (AR) @[ IoU=0.50:0.95 | area= small | maxDets=1000 ] = 0.415
Average Recall     (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=1000 ] = 0.643
Average Recall     (AR) @[ IoU=0.50:0.95 | area= large | maxDets=1000 ] = 0.682

 Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.369
Average Precision  (AP) @[ IoU=0.50      | area=   all | maxDets=1000 ] = 0.787
Average Precision  (AP) @[ IoU=0.75      | area=   all | maxDets=1000 ] = 0.307
Average Precision  (AP) @[ IoU=0.50:0.95 | area= small | maxDets=1000 ] = 0.279
Average Precision  (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=1000 ] = 0.558
Average Precision  (AP) @[ IoU=0.50:0.95 | area= large | maxDets=1000 ] = 0.610
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.481
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=300 ] = 0.481
Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=1000 ] = 0.481
Average Recall     (AR) @[ IoU=0.50:0.95 | area= small | maxDets=1000 ] = 0.387
Average Recall     (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=1000 ] = 0.624
Average Recall     (AR) @[ IoU=0.50:0.95 | area= large | maxDets=1000 ] = 0.658
```

Now, looking at the prediction, it can be told that they have updated far better than in the previous model. The figures are quite accurate.

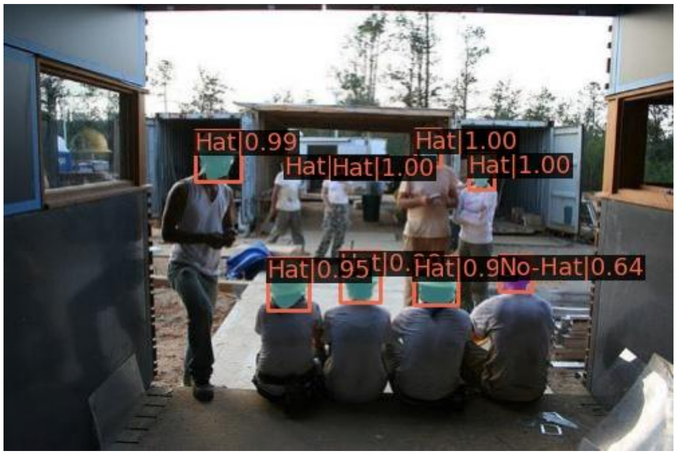


As well as the segmented mask are quite nicely settled . In the previous implementation , the masks were scattered throughout some non-target objects, but in this case that error also has been rectified.





Another modification this model has is it is only approaching the target objects.In the picture below we can see that only the people’s heads regardless of having hardhats or not are only getting detected, whereas a hardhat left inside the glass window at the left corner of the picture is not objectified, this can tell us the implementation has a far better positive result.



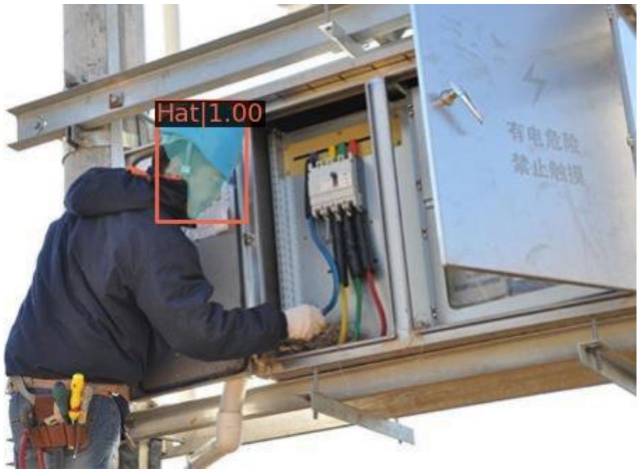
Again checking with GCnet with its syncbn feature, the results were upgraded a bit.

Scheduler: x1 (12 epochs)

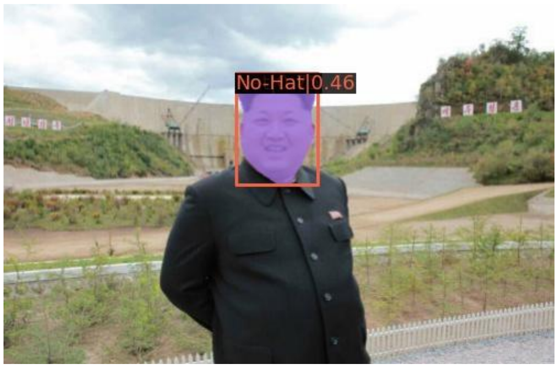
Results :

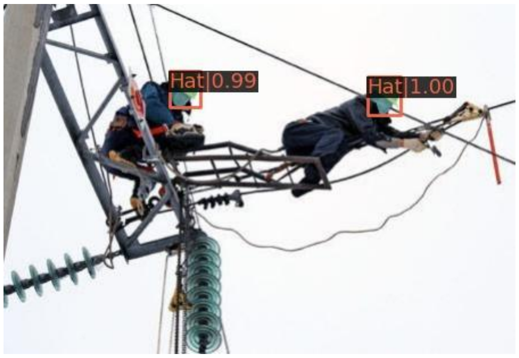
```
Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.392
Average Precision  (AP) @[ IoU=0.50      | area=   all | maxDets=1000 ] = 0.822
Average Precision  (AP) @[ IoU=0.75      | area=   all | maxDets=1000 ] = 0.318
Average Precision  (AP) @[ IoU=0.50:0.95 | area= small | maxDets=1000 ] = 0.312
Average Precision  (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=1000 ] = 0.535
Average Precision  (AP) @[ IoU=0.50:0.95 | area= large | maxDets=1000 ] = 0.579
Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.511
Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=300 ] = 0.511
Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=1000 ] = 0.511
Average Recall     (AR) @[ IoU=0.50:0.95 | area= small | maxDets=1000 ] = 0.422
Average Recall     (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=1000 ] = 0.645
Average Recall     (AR) @[ IoU=0.50:0.95 | area= large | maxDets=1000 ] = 0.709

Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.382
Average Precision  (AP) @[ IoU=0.50      | area=   all | maxDets=1000 ] = 0.817
Average Precision  (AP) @[ IoU=0.75      | area=   all | maxDets=1000 ] = 0.308
Average Precision  (AP) @[ IoU=0.50:0.95 | area= small | maxDets=1000 ] = 0.296
Average Precision  (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=1000 ] = 0.557
Average Precision  (AP) @[ IoU=0.50:0.95 | area= large | maxDets=1000 ] = 0.583
Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.492
Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=300 ] = 0.492
Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=1000 ] = 0.492
Average Recall     (AR) @[ IoU=0.50:0.95 | area= small | maxDets=1000 ] = 0.401
Average Recall     (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=1000 ] = 0.632
Average Recall     (AR) @[ IoU=0.50:0.95 | area= large | maxDets=1000 ] = 0.688
```

With this upgraded accuracy some other errors have been completely diminished.
There were no objects left to point and vice versa.



Similarly, the masks are more prominent.
Distant objects were annotated more firmly by the model which was a bit less for the DCN implementation.





Conclusion :

Mostly the objects are identified as hats, as most of the objects are hats in this dataset, thus we can explain less amount of “No-Hat” prediction and vice versa. The lowered segmentation error can be described using the similarity of the color of the hats and the jackets as an attention model is made to identify the similarity in the locality. Model training time is not that high but the model weight is too large, thus presenting itself as a research-based model so far.

Nevertheless, swin bodies were designed to catch the correct feature, but low-level models couldn’t achieve that much accuracy whereas heavy models couldn't be trained due to high requirements.

Again the deformable networks performed well in this operation and marked themselves as a high-performance model in a quite cheaper way.

THE END