## 1. Fine tuning and Results:

For this task, I have executed some variants of the MaskRCNN model, tunning its parameters to maximize the accuracy. Here are the trials and the results using the dataset_person.
Note that trial one is the baseline (trained), so the other ones are attempts to increase the precision of the model.

| | Optimizer | Scheduler | lr | Momentum | Weight decay | Epochs |
|---|---|---|---|---|---|---|
| Trial 1 | SGD | StepLR | 0.0005 | 0.9 | 0.0005 | 5 |
| Trial 2 | SGD | StepLR | 0.001 | 0.9 | 0.0005 | 5 |
| Trial 3 | Adam | StepLR | 0.001 | None | 0.0005 | 5 |
| Trial 4 | Adam | CosineAnnealingLR | 0.005 | None | 0.0005 | 5 |
| Trial 5 | SGD | StepLR | 0.001 | 0.9 | 0.005 | 5 |
| Trial 6 | SGD | StepLR | 0.001 | 0.9 | 0.0005 | 10 |

| | Average precision Bbox | Average Precision Segm | Loss in the last epoch | Loss Classifier in the last epoch |
|---|---|---|---|---|
| Trial 1 | 0.807 | 0.745 | 0.1244 | 0.0093 |
| Trial 2 | 0.813 | 0.753 | 0.1442 | 0.0111 |
| Trial 3 | 0.000 | 0.000 | 1.0211 | 0.1041 |
| Trial 5 | 0.752 | 0.703 | 0.1747 | 0.0153 |
| Trial 6 | 0.869 | 0.780 | 0.1230 | 0.0087 |

It seems that modifying the learning rate had a positive impact on the model's performance compared to the baseline. The results improved when the learning rate was increased, indicating that the initial learning rate was too low.
Additionally, I decided to switch to the Adam optimizer, which is commonly used for similar tasks, but it did not yield the desired results. The model's performance deteriorated with Adam as the optimizer. This suggests that the SGD optimizer might be more suitable for the task at hand.
Furthermore, I increased the weight decay as part of the tuning process. However, this adjustment did not result in improved performance. In fact, the larger weight decay value had a negative impact on the results. It is possible that the chosen weight decay value was too large compared to the baseline, leading to the undesired outcome.
With these adjustments in the learning rate, optimizer, and weight decay I aimed to optimize the model's performance. While some changes had positive effects, others did not yield the expected improvements or even had a negative impact on the results. So in conclusion, Trial2 resulted to be the best one.

For trial 4 I decided to stop it as at epoch 3 the loss started to have values such as: 2203978481779502743552.0000. So, obviously, I decided to not waste any more execution time on it.

This is not the only model I have tried, I trained a U-Net, a MaskRCNN with a backbone bigger than the given in the task and some more of the torch.vision module. But all of them gave me worser results than the normal ResNet.
Leaving apart the models from torch.vision, I have also discovered new ones like SAM (segment anything Model) and OneFormer, new image segmentation models released a few months ago by Meta. Both models gave incredible results, the segmentation was better (at least visually) than the one obtained with the MaskRCNN. I have not been able to adapt these models to obtain the predictions as asked due to lack of time, but I am pretty sure the results would be awesome. You can see the codes attached in the submission if you like.

## 2. Pre-Processing:

Preprocessing steps:

1. **Resizing**: The image is resized to a fixed size to ensure consistency in the input dimensions. This is necessary because the model expects inputs of a specific size. Resizing also helps to reduce computational requirements during training and inference.
2. **Normalization**: The pixel values of the image are normalized to a common scale to make them suitable for the model. This typically involves dividing the pixel values by 255 to bring them within the range of 0 to 1. Normalization helps in achieving numerical stability during training and improves convergence.
3. **Conversion to Tensor**: The image is converted into a tensor, which is the primary data format used by deep learning frameworks. The tensor representation allows for efficient computation on GPUs and integration with other model operations.
4. **Channel Ordering**: Depending on the framework and model architecture, the channel ordering of the image may need to be adjusted. For example, some models expect images to be in the "RGB" channel order (Red, Green, Blue), while others may require "BGR" (Blue, Green, Red). The channel ordering is typically handled during the conversion to tensor.

With these preprocessing steps I ensure that the input image is in a suitable format and range for the instance segmentation model.

## 3. Post-Processing:

For the post-processing I use a modified version of draw_segmentation_map. The new one, draw_segmentation_map2, takes as input the original image and the output of the instance segmentation model (target) and performs post-processing to generate the final segmentation map.
The function initializes an empty array pred with the same shape as the input image. It then iterates over the masks in _masks and assigns a unique value from 1 to len(_masks) to the pixels within each mask. This way the background will have value 0 and the masks will have different values.

Overall, this function applies thresholding, accumulates masks, adjusts mask shapes, darkens pixels outside masks, and generates a final segmentation map with unique pixel values for each object instance. The generated segmentation map can be visualized or further processed as needed.

## 4. Losses used

- **Classification Loss:** Measures discrepancy between predicted class probabilities and ground truth labels for each region of interest (ROI). Encourages accurate object classification.

- **Box Regression Loss:** Calculates difference between predicted and ground truth bounding box coordinates. Aids in refining object localization.

- **Mask Loss:** Measures similarity between predicted and ground truth masks. Encourages accurate pixel-level segmentation.

- **Objectness Loss:** Assesses how well the region proposal network discriminates between object proposals and background regions.

- **RPN Box Regression Loss:** Like box regression loss but specific to the region proposal network. Aids in refining region proposals.

## 5. Advantages and disadvantages of using global IoU

<u>**Advantages:**</u>

- **Simplicity:** it provides a simple and straightforward measure of segmentation accuracy, making it easy to interpret and compare results across different models.
- **Aggregation:** It allows for the evaluation of the model's performance across a diverse range of images by providing a consolidated measure of segmentation quality. It also considers variations in object sizes, shapes, and appearances.
- **Object-level evaluation:** is computed at the object level, considering the overlap between predicted and ground truth masks for individual objects. This allows for a fine-grained assessment of the model's ability to accurately segment each object in the image.

<u>**Disadvantages:**</u>

- **Bias towards dominant classes:** If the dataset is heavily skewed towards certain classes or objects, the metric may be biased towards those dominant classes. This means it may not adequately reflect the model's performance on less frequent or challenging classes.
- **Insensitive to localization errors:** Only evaluates the similarity of predicted and ground truth masks based on overlap, without considering the accuracy of object localization. Therefore, a model that produces well-segmented masks but with incorrect object positions may still achieve high mean IoU scores.
- **Limited context awareness:** Does not capture contextual information beyond the individual objects. It does not account for higher-level scene understanding, object relationships, or semantic consistency. As a result, it may not fully capture the model's ability to capture complex scene structures.
- **Equal weighting of images:** Assigns equal importance to each image in the test split, regardless of image difficulty or complexity. Some images may pose more challenges for segmentation due to occlusions, complex backgrounds, or object ambiguities. The mean IoU metric may not adequately reflect the model's performance on such challenging cases.