M5 Object detection

Week 3: Challenges of Object Detection and Instance Segmentation

Group 6: José Manuel López Camuñas, Marcos Conde Osorio, Alex Martín Martínez

INDEX

- Week 2: fine-tuning results
- Datasets and models
- Task A: out of context
- Task B: transplanting new objects
- Task C: qualitatively transplant
- Task D: feature inference

WEEK 2: Fine -Tuning results

As the past week we couldn't provide results on the fine-tuning of the models due to the crash of one of our PCs, we now proceed to show the results of the experiments made.

After the past week inferences, we found that the best model tested from the model_zoo from detectron2 was the R_50_FPN_1x we performed our experiments only with that model.

The parameters tested were the **learning rate** and the **batch size of the ROI head,** as this were the ones that would affect the most to the performance.

For the experiments on the learning rate, the **batch size of the ROI head** for the detection and segmentation was 512 and 64 respectively.

WEEK 2: Fine -Tuning results

Faster R-CNN:
R_50_FPN_1x

Learning rate	mAP
1e-3	58.038
1e-4	55.942
1e-5	14.856

Mask R-CNN:
R_50_FPN_1x

Learning rate	mAP detection	mAP segmentation
1e-3	58.384	45.572
1e-4	52.938	43.370
1e-5	27.39	26.838

For both models we trained for 2000 iteration and we can observe that a higher learning rate makes the models perform better. We also tried with a higher learning rate but in this case the model mAP decreases which indicates that 0.001 is the optimal learning rate.

WEEK 2: Fine -Tuning results

Faster R-CNN:
R_50_FPN_1x

ROI batch size	mAP
256	57.160
512	58.038
1024	55.470

Mask R-CNN:
R_50_FPN_1x

ROI batch size	mAP detection	mAP segmentation
64	58.384	45.572
124	57.239	46.363
256	56.064	44.988

Testing with different ROI batch sizes we can observe that the models differ. For the Faster R-CNN we have an optimal batch size of 512 while for Mask R-CNN depending on the task we should be using a batch size of 64 or 124. The ROI batch size tried in the models differ because in the case of the Mask R-CNN would give us an OOM error.

WEEK 2: Fine -Tuning results

Final comparision: detection

Faster R-CNN

	AP	AP50	AP75	APs	APm	APl	Person AP	Car AP
Pre-trained	43.76	76.15	44.21	26.64	53.19	62.70	28.77	58.75
Fine-tuned	58.04	82.24	67.44	32.38	64.00	71.04	52.18	63.69

Mask R-CNN

	AP	AP50	AP75	APs	APm	APl	Person AP	Car AP
Pre-trained	54.81	79.26	62.11	39.70	65.06	62.81	44.48	65.14
Fine-tuned	58.38	80.52	67.88	41.78	68.85	74.94	51.14	65.63

WEEK 2: Fine -Tuning results

Final comparision: segmentation

Mask R-CNN

	AP	AP50	AP75	APs	APm	APl	Person AP	Car AP
Pre-trained	43.76	76.15	44.21	26.64	53.19	62.70	28.77	58.75
Fine-tuned	45.5	77.03	47.42	28.58	55.85	69.56	31.11	60.02

For both models we can observe a significative increase in the AP for the detection task as well as in the person class AP which was the one that gave more troubles. For the segmentation task we can see that the impact is not as noticeable as in the detection but the fine-tuning still enhances the model performance.

Datasets and models

For this week we used images from two different datasets to perform the experiments:

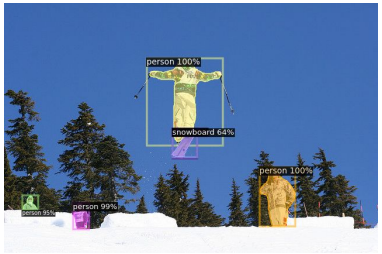
- Out of context dataset
- COCO validation dataset

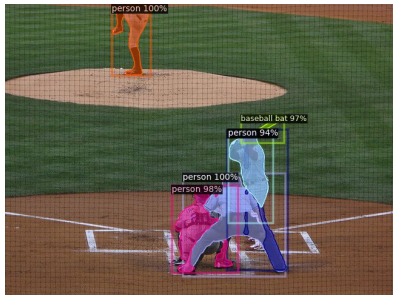
The objective of performing inference on this datasets is to see how the models behave when changing features of the instances being detected or segmented. To analyze this behaviour we performed modifications on the images of COCO datasets that the models don't expect and we will comment the most remarkable results.

The models used to perform the detection and the segmentation will be the Faster R-CNN and the Mask R-CNN with the R_50_FPN_1x from the model_zoo from detectron2.

Datasets and models

Here we show some of the original images that we used from the COCO dataset and how the model behaves with them.

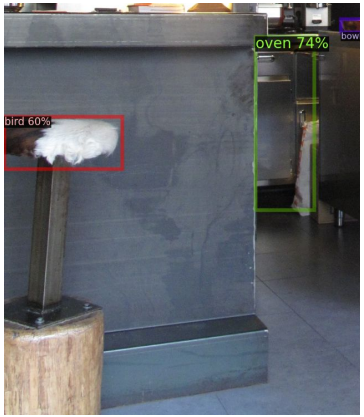


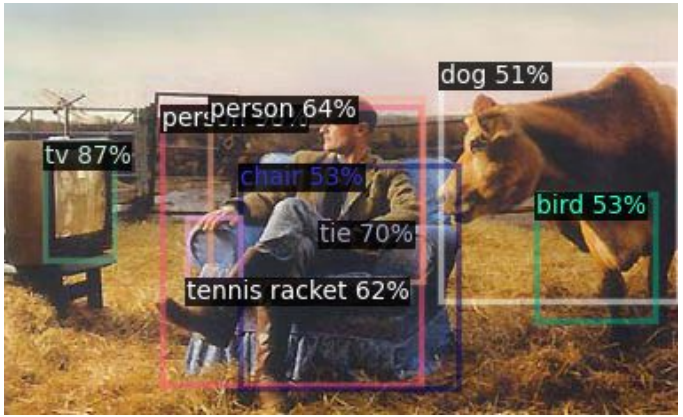


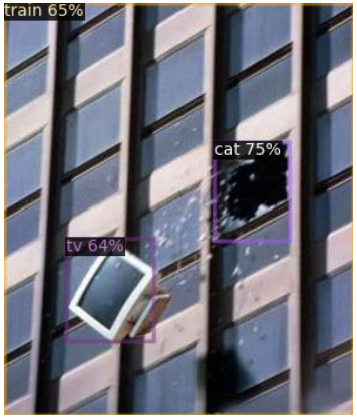
Task A: out of context

From the **Out of context** split we obtained the following

Faster R-CNN







First we can see that the model detects the chair as a bird as it is lined with what seems feather. The next photo we can see how it detects a tennis racket as it can be seen something that haves the shape of a racket under a hand. The last one probably is detecting a train because of the windows of the building.

Task A: out of context

From the **Out of context** split we obtained the following

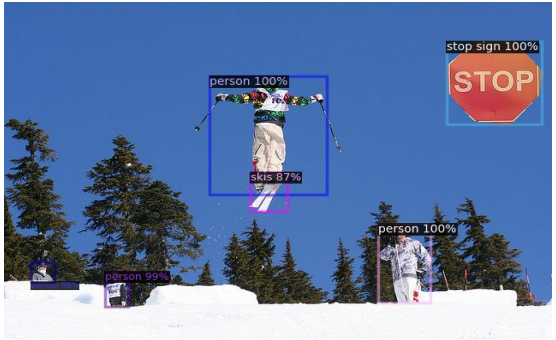
Mask R-CNN

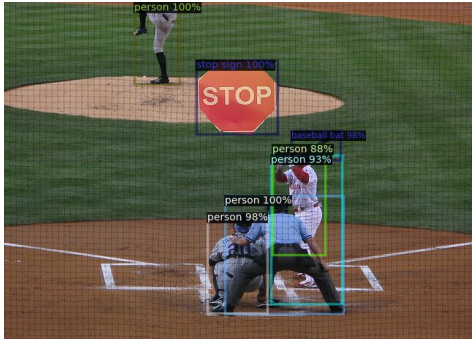
For this model we observed that it detected less objects that in the other case but the detected objects were well segmented. We can see in the photo in the middle that detects the shoe as an airplane most probably because of the size and the color of it. One very random prediction is the bear with a snowboard when it is in a puddle.

Task B: transplanting new objects

In this case we randomly transplant objects into other images. We transplanted **stop signals, microwaves and toasters.**

Faster R-CNN: Stop signals

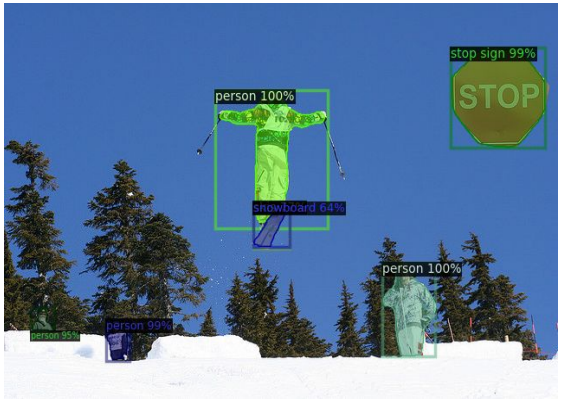


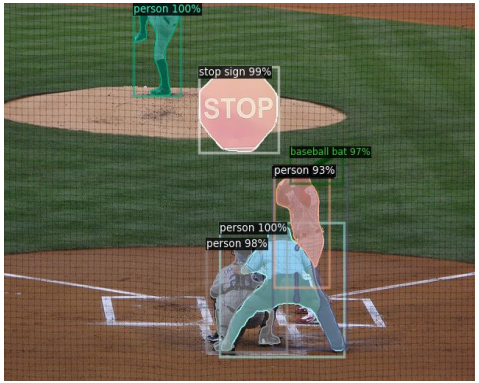


For the stop sign we had that it was well detected in any position and image that it was inserted probably because the red color and that it was a big signal.

Task B: transplanting new objects

Mask R-CNN: Stop signals

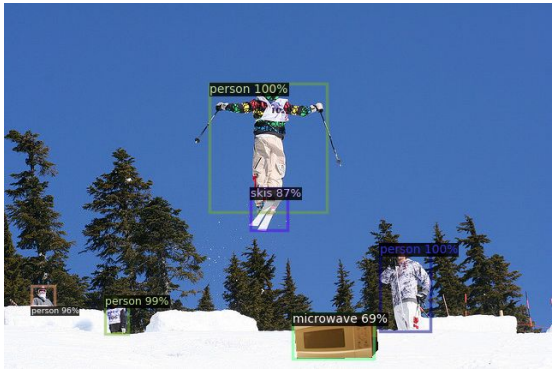




We can see the same as with the Faster R-CNN

Task B: transplanting new objects

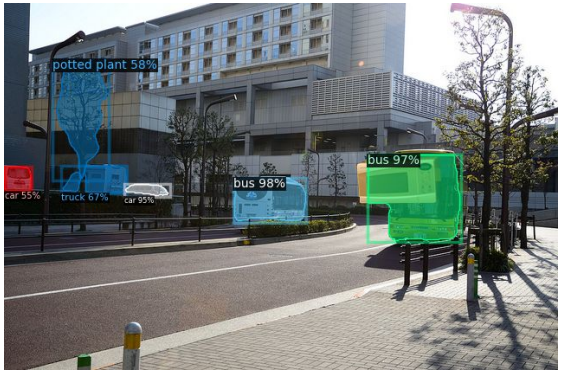
Faster R-CNN: Microwaves

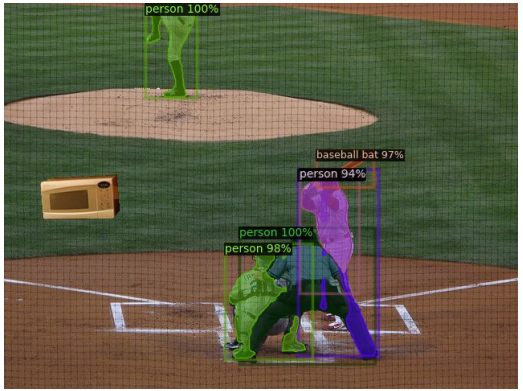


With the microwave we can see that the confidence is lower than when the microwave is situated in a kitchen and in the right image we can see that the bounding box of the bus is deformated because of the microwave as if it was part of the bus.

Task B: transplanting new objects

Mask R-CNN: Microwaves

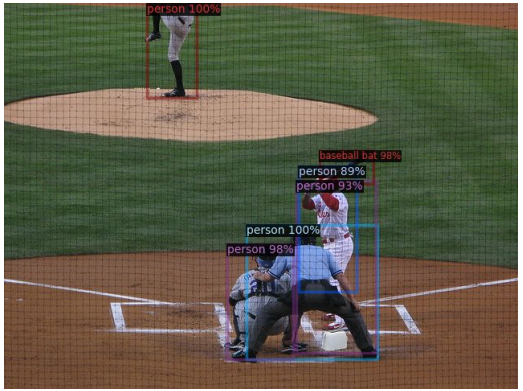




With the Mask R-CNN model we can see also the same behaviour. In the center image the bus is segmented as if the microwave was part of it probably because it haves very similar colors and texture. In the left image we can see that the color of the microwave is also present in the background difficulties the model to detect the microwave.

Task B: transplanting new objects

Faster R-CNN: Toaster



In the case of the toaster the model had a lot of trouble to detect it in other images but in the case of the image in the left we could manage to trick the model by locating it at a similar spot than the kite.

Task B: transplanting new objects

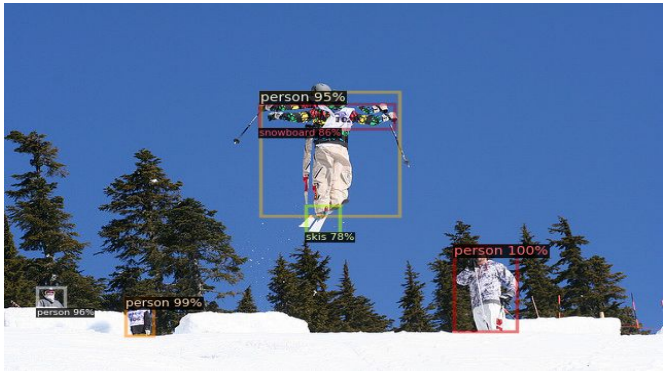
Mask R-CNN: Toaster

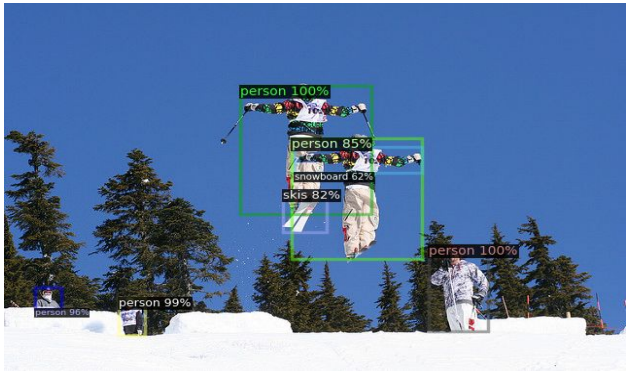
With the Mask R-CNN had also the same behaviour

Task C: qualitatively transplant

In this task we transplanted one object of the image to another position.

Faster R-CNN

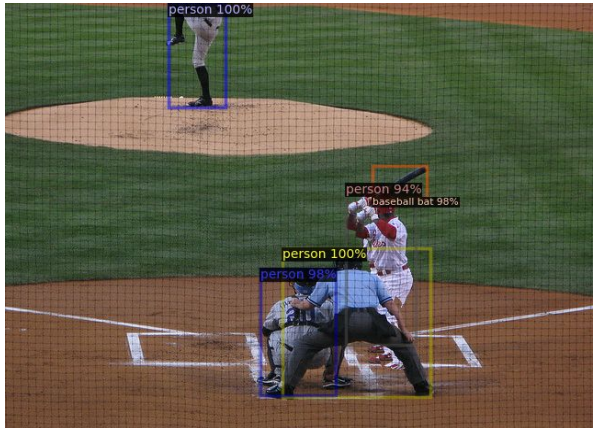


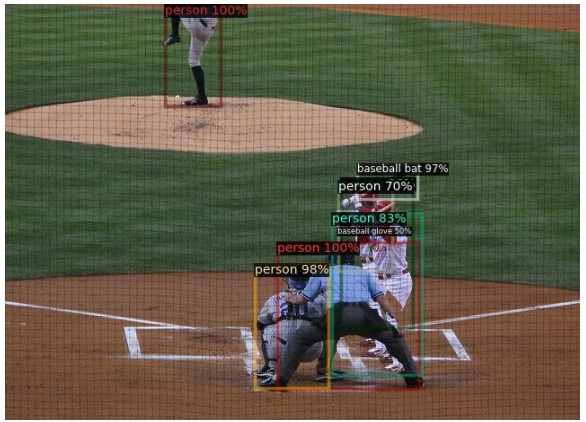


In this example we can see that when an object is repeated the detected objects may change. At the image on the left repeating the jumping skier made the model correctly detect the skis but also detected the arms of the skier as a snowboard but still detecting the person. when separated enough another person is also detected but the arms still seem to be a snowboard for the model.

Task C: qualitatively transplant

Faster R-CNN





In this case the repetition of the baseball player doesn’t affect negatively the model it caused in fact the model to find other object like the baseball glove.

Task C: qualitatively transplant

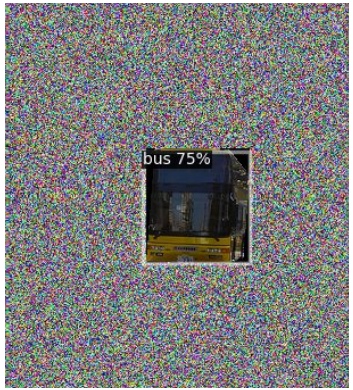
Mask R-CNN

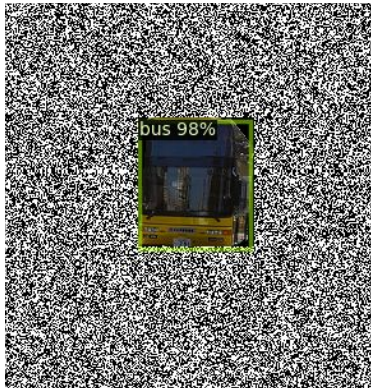
For the mask R-CNN model we didn’t found many remarkable results apart from this one where the repetition of the boat enables the model to detect the boat in the image but the repetition is detected as a truck probably because of the location and the size that could fit as there are also persons in this height of the image.

Task D: feature inference

Here we tried to perform modification on the images leaving only 1 of the objects and see how it affected the performance.

Faster R-CNN

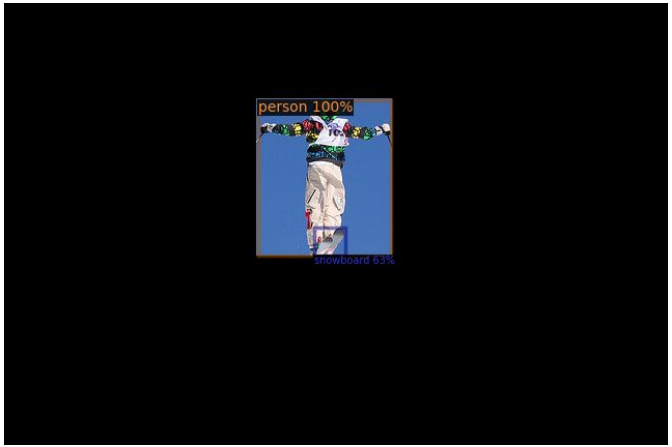


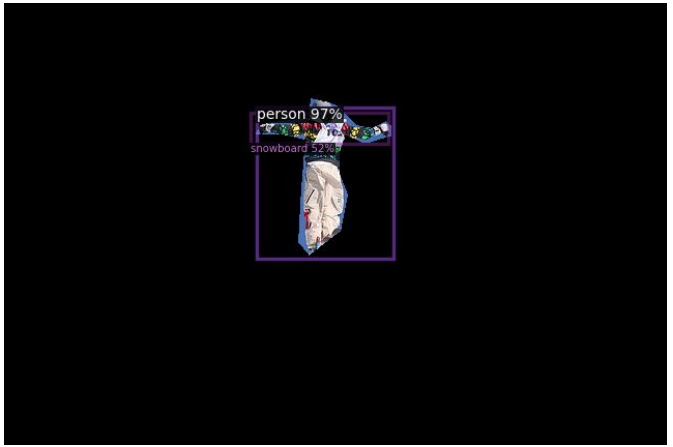


Changing the background of the image with white noise affects the confidence of the model to classify correctly the images. In the case of black and white noise the effect was not strong.

Task D: feature inference

Faster R-CNN

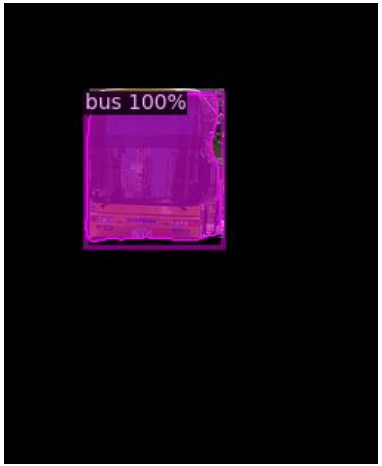


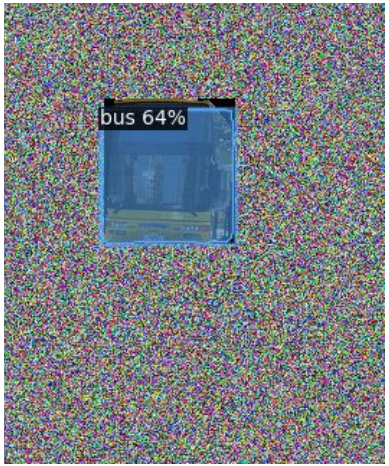


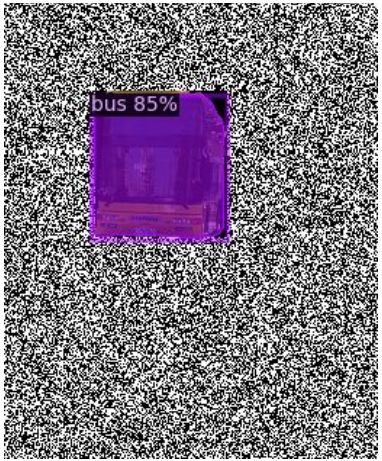
With the skier we observe the person and the skies are correctly detected when leaving the bounding box background but when removing it the model starts to detect another time the arms as a snowboard, probably because of the colors of the jacket.

Task D: feature inference

Mask R-CNN



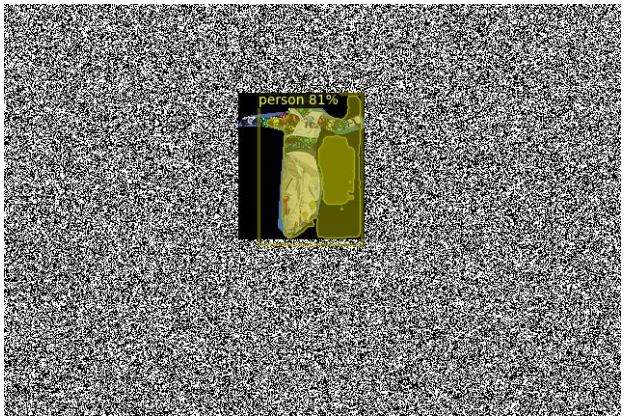


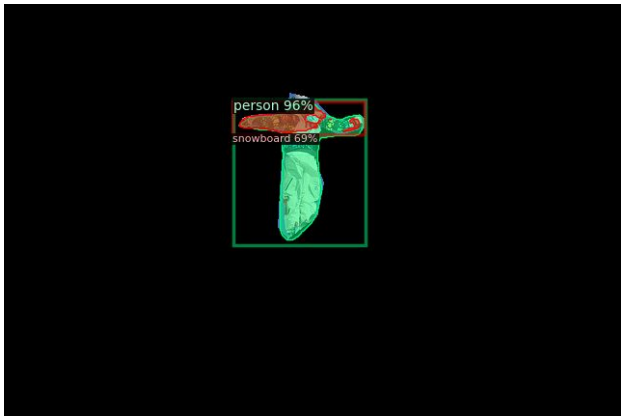


With the Mask R-CNN model we observe the same. The white noise affects the connfidence detection as well as the mask that it detects.

Task D: feature inference

Mask R-CNN





In this case we can see that the black and white noise had more effect than in the other cases making the model to predict the mask with a very strange shape for a person but with a high confidence. On the right we can see how the model also detects the snowboard in the skier arms.

Final summary

Task a)
The out of context images forced the model to have unexpected behaviours but overall could perform better than expected.

Task b)
When transplanting objects at another image we mostly saw that the models had trouble to detect them if there was some similar texture or colors in the destination image or could wrongly classify the object. If the object is not similar to anything in the image then it can correctly detected most of the times.

Task c)
When transplanting an object over the image we found that in many cases it actually helped the model to correctly detect other objects which in the original image weren't detected, although in general it was less accurate.

Task d)
When the object are isolated from the rest of the image the model doesn't seem to have strange behaviours but when adding noise at the background we can observe a worse performance of the models.