Gedeon Muhawenayo
gmuhawenayo@aimsammi.org
African Masters in Machine Intelligence
Computer Vision II

LAB1: Detectron2

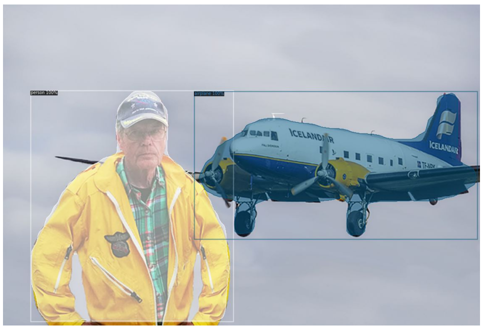
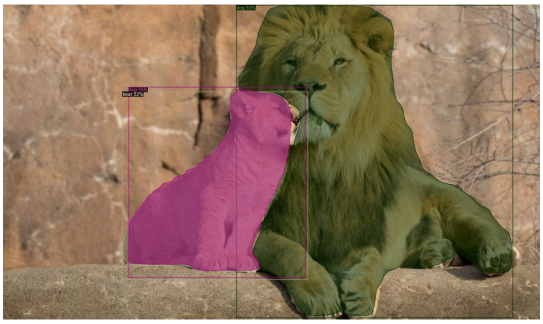
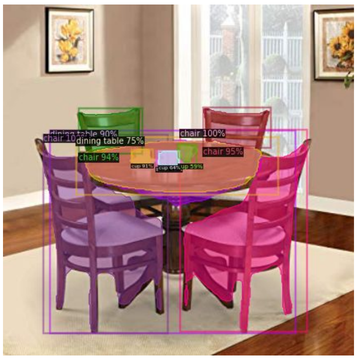
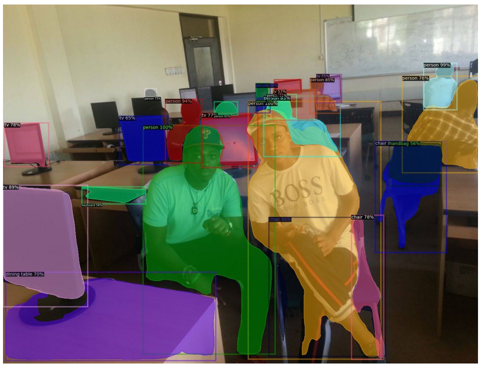
Summary:

In this LAB, I got to familiarize myself with Detectron2 object detection library. I started by reading documentation regarding required dependencies to be able to install and to use Detectron2. Next, I have explored pretrained models for semantic segmentation and pose estimation. This LAB has helped me to clearly understand Lecture1 and has widened my thinking about object detection and segmentation tasks as they related to practical applications and how they could be used to tackle social problems.

Part B: Running pretrained model for instance segmentation

Model architecture: From Detectron2 model zoo I have used the default predictor of model configuration file named `COCO-InstanceSegmentation/mask rcnn R 50 FPN 3x.yaml` Which is a pretrained instance segmentation baseline with **Mask R-CNN.** It was pre-trained on **COCO-dataset** with a backbone of **ResNet50.**

Examples:

Correct prediction	Incorrect prediction
	
 dining table 90% chair 10 dining table 75% chair 100% chair 94% chair 95% cup 91% cup 64% up 59%	
In these two images the model made the correct prediction because all the categories that it is aware of are few and clear.	In the first image the model made the wrong prediction because the dog looks similar to the lion, but in our dataset we do not have a lion. In the second image model made mistakes because we have too many instances and desktop computers look like TV.

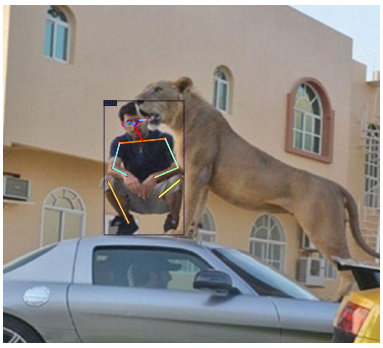
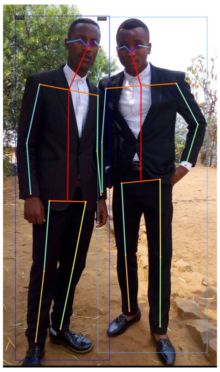
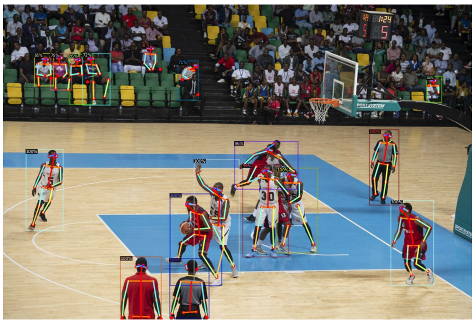
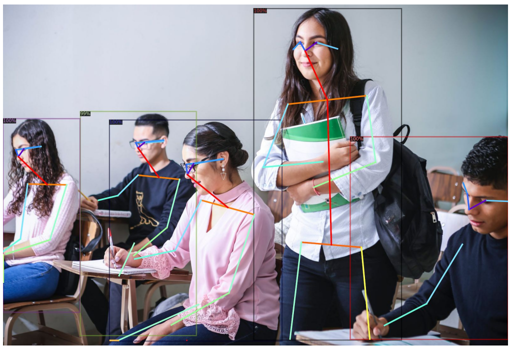
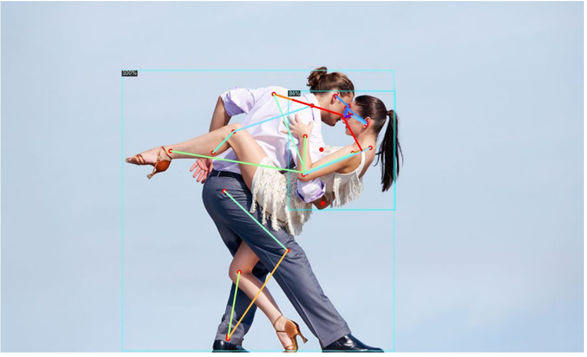
Observations: If the image is clear and does not have too many instances Detectron2 will make perfect predictions. The model seems to be more accurate on some objects than the others (eg: Airplane, Person).

Error mode: For the model of instance segmentation I have realized the error of failing to segment objets when we have too many instances within the input image. It keeps predicting lion, tigers as dogs. It also keeps confusing tv and Desktop. When objects are somehow occluded, It makes two bounding boxes on a single object, predicting that object as two different categories.

Part B: Running pre trained model for pose estimationModel architecture:

From Detectron2 model zoo I have used the default predictor of model configuration file named `COCO-Keypoints/keypoint_rcnn_R_50_FPN_3x.yaml` Which is a pretrained keypoint detection model baseline. It was pre-trained on **COCO-dataset** with a backbone of **ResNet50.**

Examples:

Correct prediction	Incorrect prediction
 	
	
For a simple and clear image the model will make correct predictions. This depends on the pose of the detected person and the number of instances.	The model will most make incorrect predictions if the image has too many instances(image1) and if the pose of the person detected is complicated(image2).

Observations: If the image is clear, with few instances and the poses of the detected person are not very complex, the model will be able to make good pose estimation. The predicted pose is not 100% perfect sometimes there is a small shift or angle difference with the perfect pose. The model does not estimate the pose for objects other than person at all.

Error mode: The model fails to estimate pose for people who have some part of their body being occluded by other people. It also makes wrong predictions when estimating pose for the legs of babies. It makes wrong predictions for images of a big group of people and it will fail to totally estimate the pose of some of them.