University of North Carolina at Charlotte
Department of Electrical and Computer Engineering

ECGR 8119

Applied Artificial Intelligence (AI)

Report On

Building a real-time Mask R-CNN using Detectron

Prepared by:
Md Tareq Mahmud
ID: 801311662

Submitted To:
Dr. Minhaj Nur Alam
Assistant Professor
Dept. of ECE

GitHub repository: https://github.com/Tareq-BD/ECGR-8119-Applied-AI

This project is about building a real-time Mask R-CNN using Detectron. Detectorn2 is the latest Python library for object detection released by the AI Facebook researchers' team. I have used this repository. The main task was to implement a simple solution to run Detectron Mask R-CNN algorithm for object detection and instance segmentation with webcam. In coding section, I have maintained the following sequential steps:

1. Installation Dependencies and Libraries:

```
!python -m pip install pyyaml==5.1
import sys, os, distutils.core
# Note: This is a faster way to install detectron2 in Colab, but it does not include all functionalities
# See https://detectron2.readthedocs.io/tutorials/install.html for full installation instructions
!git clone 'https://github.com/facebookresearch/detectron2'
dist = distutils.core.run_setup("./detectron2/setup.py")
!python -m pip install {' '.join([f"'{x}'" for x in dist.install_requires])}
sys.path.insert(0, os.path.abspath('./detectron2'))

# Properly install detectron2. (Please do not install twice in both ways)
# !python -m pip install 'git+https://github.com/facebookresearch/detectron2.git'
```

2. Examining the dependencies (pytorch, cuda and detectron2 version)

```
[ ] import torch, detectron2
    !nvcc --version
    TORCH_VERSION = ".".join(torch.__version__.split(".")[:2])
    CUDA_VERSION = torch.__version__.split("+")[-1]
    print("torch: ", TORCH_VERSION, "; cuda: ", CUDA_VERSION)
    print("detectron2:", detectron2.__version__)
```

```
nvcc: NVIDIA (R) Cuda compiler driver
Copyright (c) 2005-2021 NVIDIA Corporation
Built on Sun_Feb_14_21:12:58_PST_2021
Cuda compilation tools, release 11.2, V11.2.152
Build cuda_11.2.r11.2/compiler.29618528_0
torch:  1.12 ; cuda:  cu113
detectron2: 0.6
```

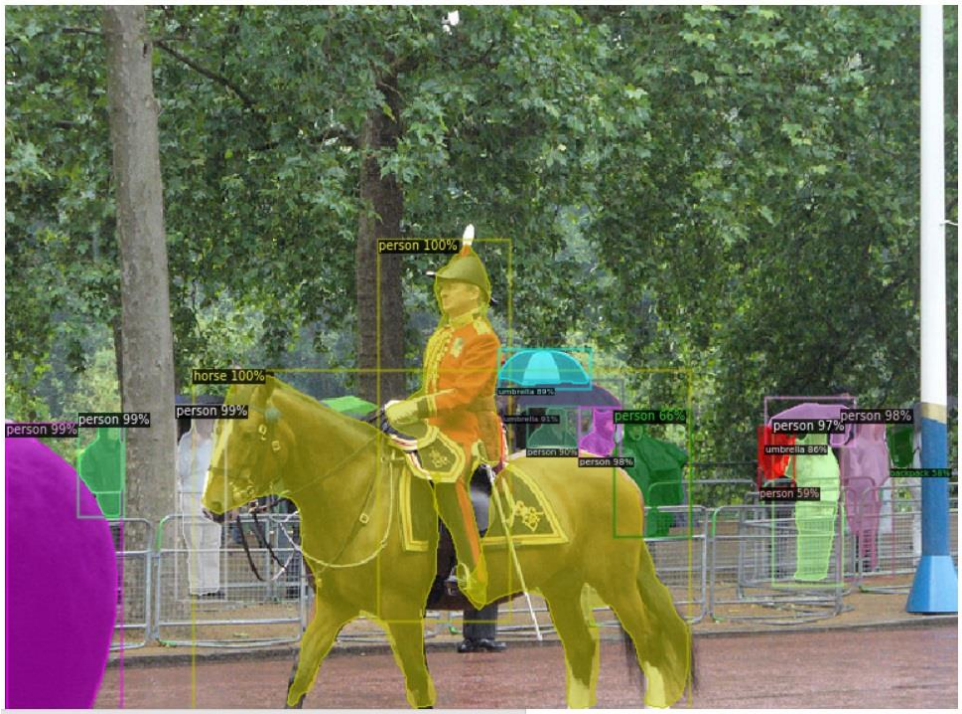
3. Adding Libraries:

```
[ ] # Some basic setup:
    # Setup detectron2 logger
    import detectron2
    from detectron2.utils.logger import setup_logger
    setup_logger()

    # import some common libraries
    import numpy as np
    import os, json, cv2, random
    from google.colab.patches import cv2_imshow

    # import some common detectron2 utilities
    from detectron2 import model_zoo
    from detectron2.engine import DefaultPredictor
    from detectron2.config import get_cfg
    from detectron2.utils.visualizer import Visualizer
    from detectron2.data import MetadataCatalog, DatasetCatalog
```

4. Running a pre-trained model:
   In this step I have used model_zoo which is pretrained on the COCO dataset. First an image was download from the COCO dataset then, a detectron2 ***config*** and a detectron2 ***DefaultPredictor*** was created to run inference on this image. Finally, got the segmented image output.

Input Image

Segmented Image



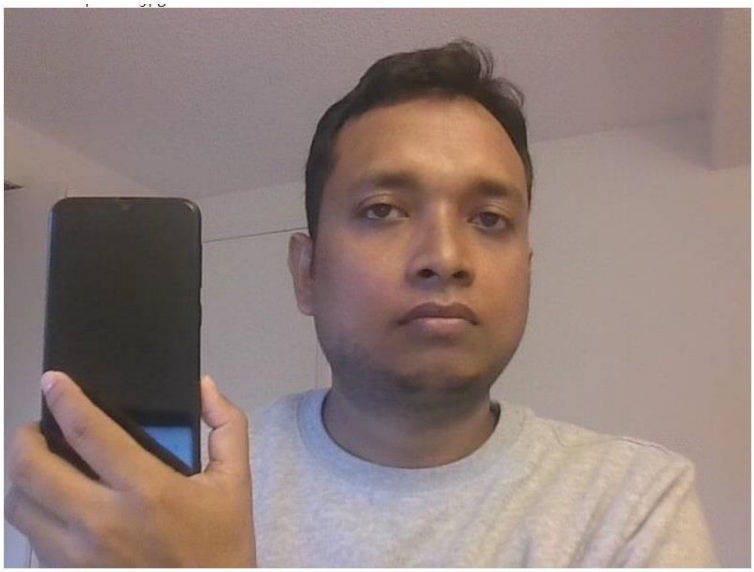
For object detection and instance segmentation I have used three different strategies are as follows:

3. Object detection and instance segmentation from image captured by webcam.
4. Object detection and instance segmentation from realtime video captured by webcam.
5. Object detection and instance segmentation from youtube video.

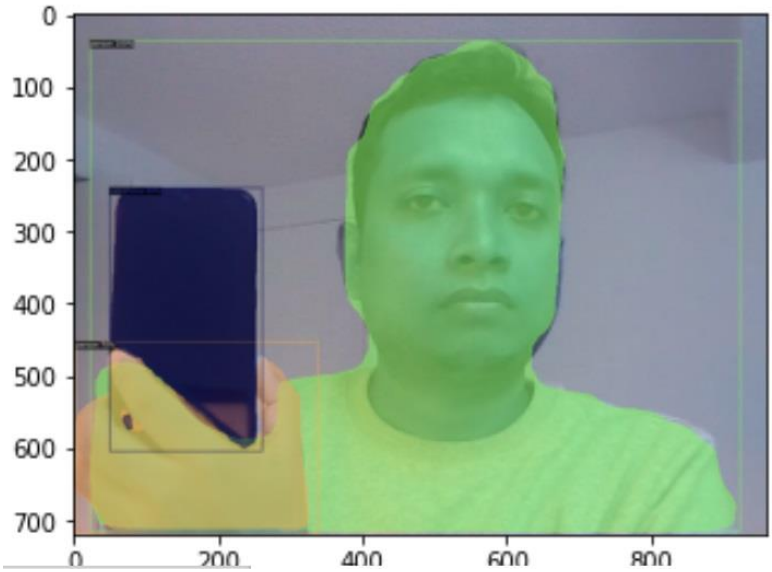
Object detection and instance segmentation from image captured by webcam:

I have taken a picture from webcam and the output picture contains the object detection and instance segmentation.

Input Image



Segmented Image



Object detection and instance segmentation from realtime video captured by webcam:

A realtime video is captured by my laptop webcam and provides object detection with instance segmentation. However, the output video is very slow.

Object detection and instance segmentation from youtube video:

In this step, I have used a youtube video as input and get the output video with object detection with instance segmentation.

YouTube video Link: https://www.youtube.com/watch?v=1XIOmKGjgho

Link of the output video: https://github.com/Tareq-BD/ECGR-8119-Applied-AI/blob/main/video-output.mkv