Mask_RCNN official library - https://github.com/matterport/Mask_RCN (https://github.com/matterport/Mask_RC

Faster_RCNN detectron2 - https://github.com/facebookresearch/de (https://github.com/facebookresearch/de

Coco Dataset - https://cocodataset.org/ (https://cocodataset.org/)

Step #1 - Extract Person Mask

Import Libraries

In [1]:
```
import sys
import random
import math
import os
import cv2
import numpy as np
import skimage.io
import matplotlib
import matplotlib.pyplot as plt

from mrcnn.config import Config
from mrcnn import utils
import mrcnn.model as modellib
from mrcnn import visualize

# Root directory of the project
ROOT_DIR = os.path.abspath("../")
sys.path.append(os.path.join(ROOT_DIR, "codes/utills/coco/"
import coco
from pycocotools.coco import COCO
```

```
Using TensorFlow backend.
```

Configuration

In [2]:
```
class_names = ['BG', 'person', 'bicycle', 'car', 'motorcycl
               'bus', 'train', 'truck', 'boat', 'traffic li
               'fire hydrant', 'stop sign', 'parking meter'
               'cat', 'dog', 'horse', 'sheep', 'cow', 'elep
               'zebra', 'giraffe', 'backpack', 'umbrella',
               'suitcase', 'frisbee', 'skis', 'snowboard',
               'kite', 'baseball bat', 'baseball glove', 's
               'surfboard', 'tennis racket', 'bottle', 'win
               'fork', 'knife', 'spoon', 'bowl', 'banana',
               'sandwich', 'orange', 'broccoli', 'carrot',
               'donut', 'cake', 'chair', 'couch', 'potted p
               'dining table', 'toilet', 'tv', 'laptop', 'm
               'keyboard', 'cell phone', 'microwave', 'oven
               'sink', 'refrigerator', 'book', 'clock', 'va
               'teddy bear', 'hair drier', 'toothbrush']
```

In [3]:
```
class InferenceConfig(coco.CocoConfig):
    GPU_COUNT = 1
    IMAGES_PER_GPU = 1

config = InferenceConfig()
# config.display()
```

Model with pretrained weights

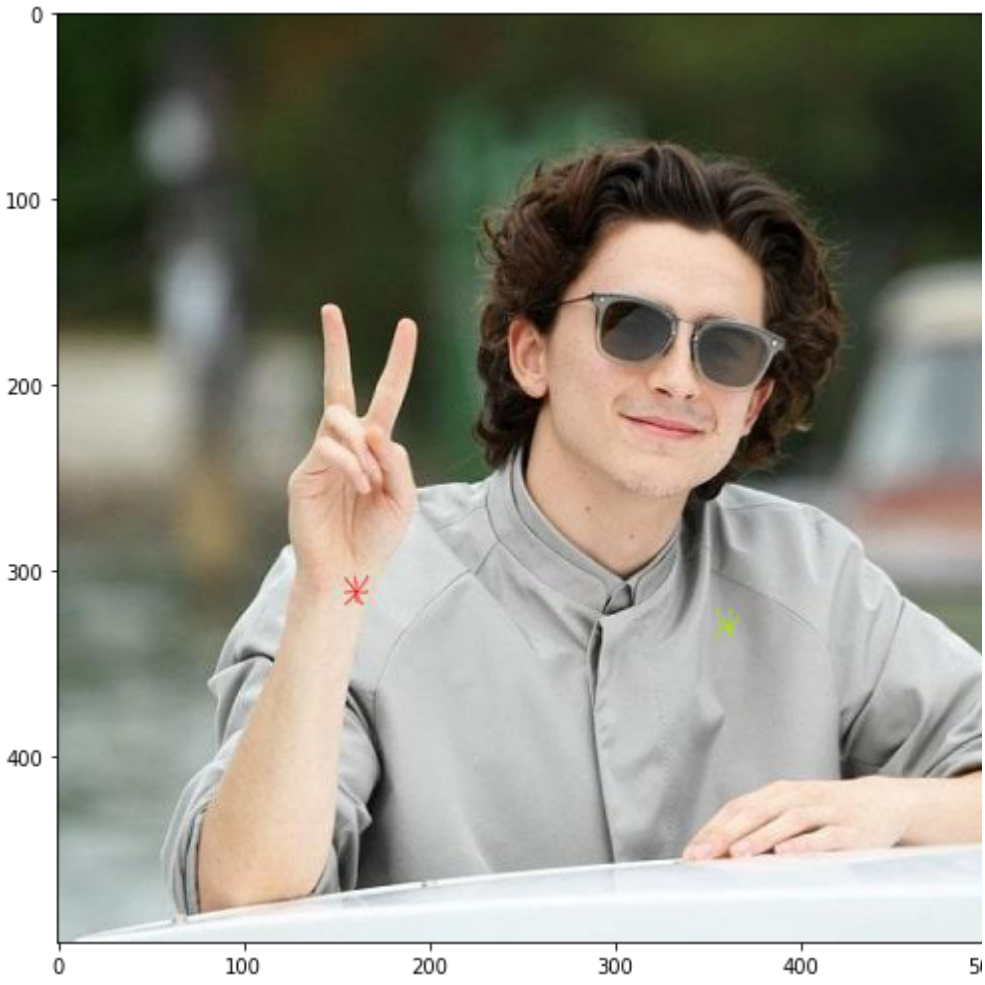
In [4]:
```
# Local path to trained weights file
COCO_MODEL_PATH = os.path.join(ROOT_DIR, "models/pretrained

# Download COCO trained weights from Releases if needed
if not os.path.exists(COCO_MODEL_PATH):
    utils.download_trained_weights(COCO_MODEL_PATH)

model = modellib.MaskRCNN(mode="inference", config=config,
# Load weights trained on MS-COCO
from keras.engine import saving
model.load_weights(COCO_MODEL_PATH, by_name=True)
if not os.path.exists(COCO_MODEL_PATH):
    utils.download_trained_weights(COCO_MODEL_PATH)
```

```
WARNING:tensorflow:From /home/farhat/Farhat_files/Envs/mark
ython3.6/site-packages/tensorflow_core/python/ops/resource_
s.py:1630: calling BaseResourceVariable.__init__ (from tens
on.ops.resource_variable_ops) with constraint is deprecated
e removed in a future version.
Instructions for updating:
If using Keras pass *_constraint arguments to layers.
WARNING:tensorflow:From /home/farhat/Farhat_files/Envs/mark
ython3.6/site-packages/keras/backend/tensorflow_backend.py:
ame tf.nn.max_pool is deprecated. Please use tf.nn.max_pool

WARNING:tensorflow:From /home/farhat/Farhat_files/Envs/mark
ython3.6/site-packages/mask_rcnn-2.1-py3.6.egg/mrcnn/model.
name tf.log is deprecated. Please use tf.math.log instead.

WARNING:tensorflow:From /home/farhat/Farhat_files/Envs/mark
ython3.6/site-packages/mask_rcnn-2.1-py3.6.egg/mrcnn/model.
re (from tensorflow.python.ops.array_ops) is deprecated and
moved in a future version.
Instructions for updating:
Use tf.where in 2.0, which has the same broadcast rule as n
WARNING:tensorflow:From /home/farhat/Farhat_files/Envs/mark
ython3.6/site-packages/mask_rcnn-2.1-py3.6.egg/mrcnn/model.
ling crop_and_resize_v1 (from tensorflow.python.ops.image_o
th box_ind is deprecated and will be removed in a future ve
Instructions for updating:
box_ind is deprecated, use box_indices instead
WARNING:tensorflow:From /home/farhat/Farhat_files/Envs/mark
ython3.6/site-packages/mask_rcnn-2.1-py3.6.egg/mrcnn/model.
name tf.sets.set_intersection is deprecated. Please use tf.
ection instead.

WARNING:tensorflow:From /home/farhat/Farhat_files/Envs/mark
ython3.6/site-packages/mask_rcnn-2.1-py3.6.egg/mrcnn/model.
name tf.sparse_tensor_to_dense is deprecated. Please use tf
dense instead.

WARNING:tensorflow:From /home/farhat/Farhat_files/Envs/mark
ython3.6/site-packages/mask_rcnn-2.1-py3.6.egg/mrcnn/model.
float (from tensorflow.python.ops.math_ops) is deprecated a
removed in a future version.
Instructions for updating:
Use `tf.cast` instead.
```

Loading an image

In [12]:
```
# Load a random image from the images folder
IMAGE_DIR = ROOT_DIR + "/data/random_images/mark_added/"
file_name = random.choice(os.listdir(IMAGE_DIR))
file_name = "random.jpg"
image = skimage.io.imread(os.path.join(IMAGE_DIR, file_name

import matplotlib.pyplot as plt
plt.figure(figsize=(10,10))
plt.imshow(image)
```

Out[12]:
```
<matplotlib.image.AxesImage at 0x7f27beeaed68>
```

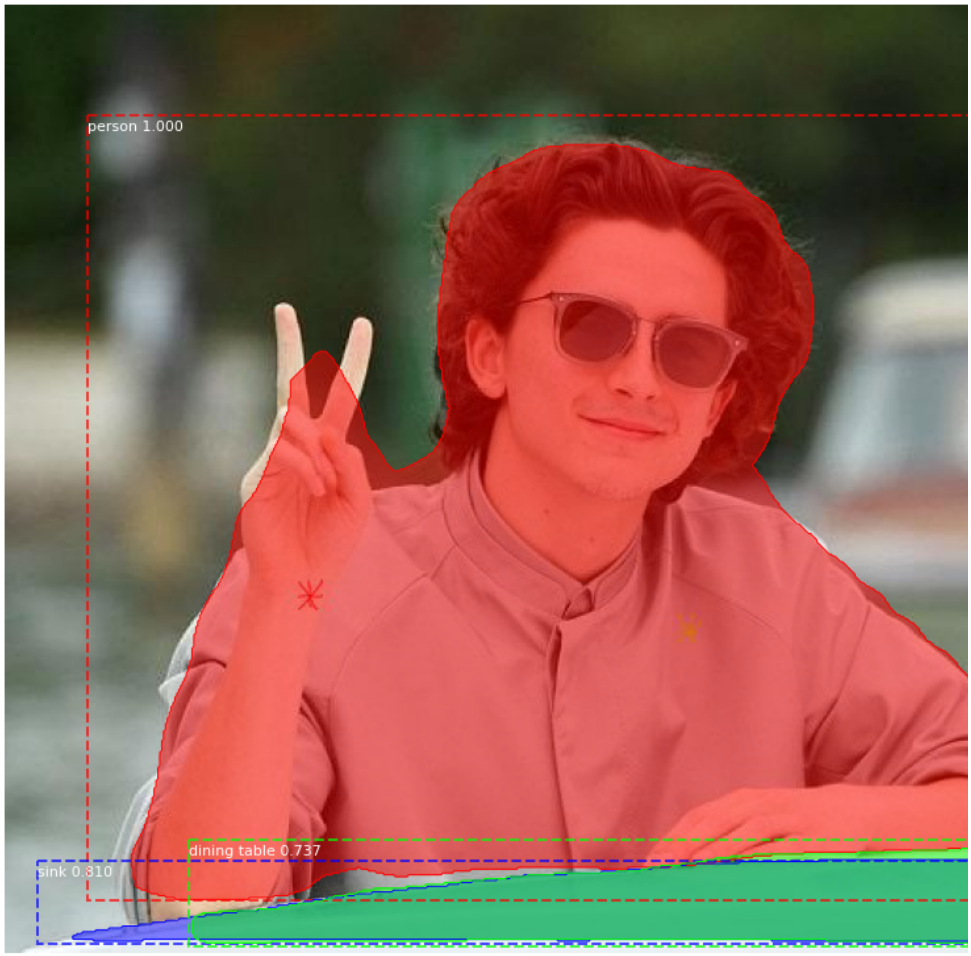


Prediction

In [13]:

```
# Run detection
results = model.detect([image], verbose=1)

# Visualize results
r = results[0]
visualize.display_instances(image, r['rois'], r['masks'], r
                            class_names, r['scores'])
from collections import Counter
print(f'Total detected objects - {len(r["class_ids"])}')
print(f'Total unique objects - {len(np.unique(r["class_ids"
print("---------------------------------")
bla = [print(f'{class_names[id]} - {num}') for id,num in Co
```

```
Processing 1 images
image                    shape: (500, 600, 3)         min:
max:  255.00000  uint8
molded_images            shape: (1, 1024, 1024, 3)    min:
max:  151.10000  float64
image_metas              shape: (1, 93)               min:
max: 1024.00000  float64
anchors                  shape: (1, 261888, 4)        min:
max:    1.29134  float32
```

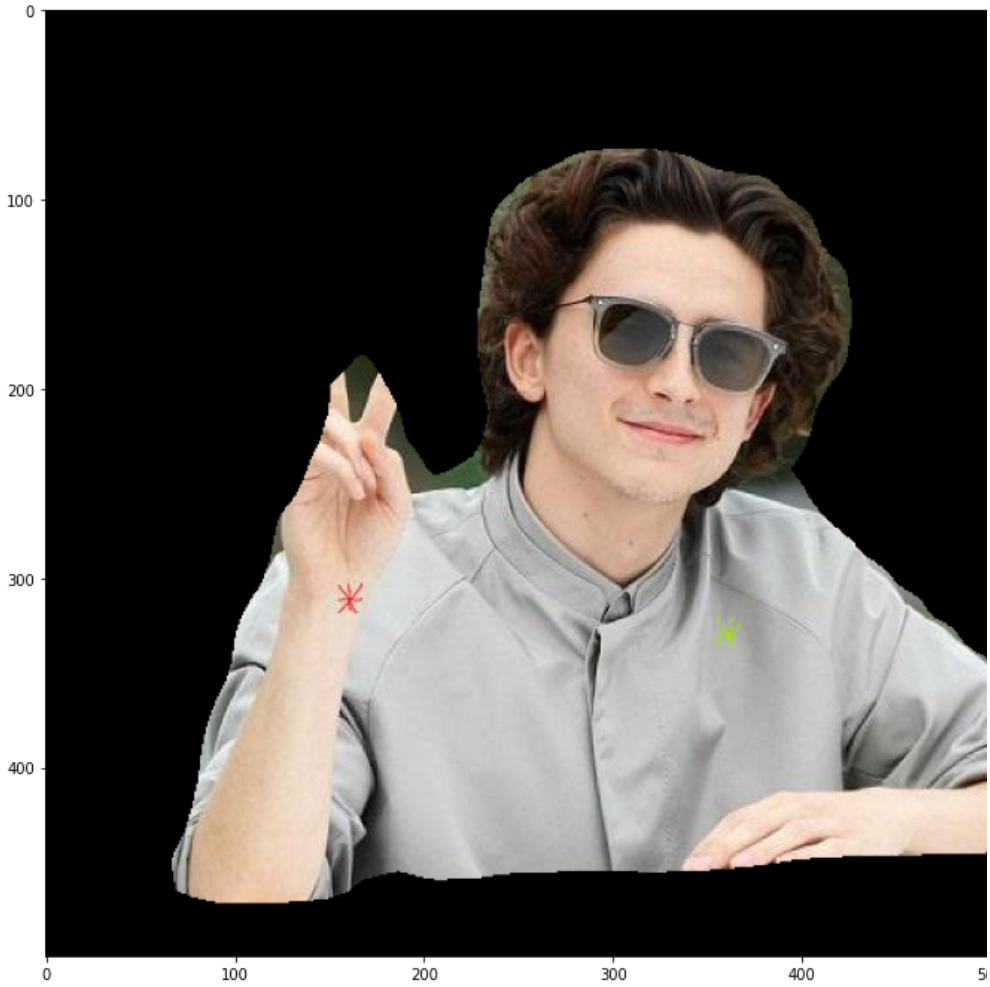


```
Total detected objects - 3
Total unique objects - 3
--------------------------------
person - 1
sink - 1
dining table - 1
```

Showing Masks

In [14]:
```
import matplotlib.pyplot as plt
ids = np.where(r['class_ids']==class_names.index("person"))

import shutil
from datetime import datetime
out_dir = os.path.join(ROOT_DIR, "outputs/mask/new/")
if not os.path.exists(out_dir): os.makedirs(out_dir)
else:
    shutil.rmtree(out_dir)
    os.makedirs(out_dir)

plt.figure(figsize=(30,30))
columns = 2
person_mask = []
for i, id in enumerate(ids):
    mask = r['masks'][:, :, id] * 1
    mask =np.moveaxis(np.stack([mask, mask, mask]), 0, 2)
    masked_image = image * mask
    person_mask.append(np.array(masked_image))
    plt.subplot(len(ids) / columns + 1, columns, i + 1)
    plt.imshow(masked_image)
    skimage.io.imsave(f'{out_dir}/{i+1}_{class_names[id]}.j
```

```
/home/farhat/Farhat_files/Envs/mark_c3.6/lib/python3.6/site
pykernel_launcher.py:20: MatplotlibDeprecationWarning: Pass
egers as three-element position specification is deprecated
and will be removed two minor releases later.
/home/farhat/Farhat_files/Envs/mark_c3.6/lib/python3.6/site
pykernel_launcher.py:22: UserWarning: /media/farhat/Researc
rk-NET/outputs/mask/new//1_BG.jpg is a low contrast image
Lossy conversion from int64 to uint8. Range [0, 255]. Conve
uint8 prior to saving to suppress this warning.
```



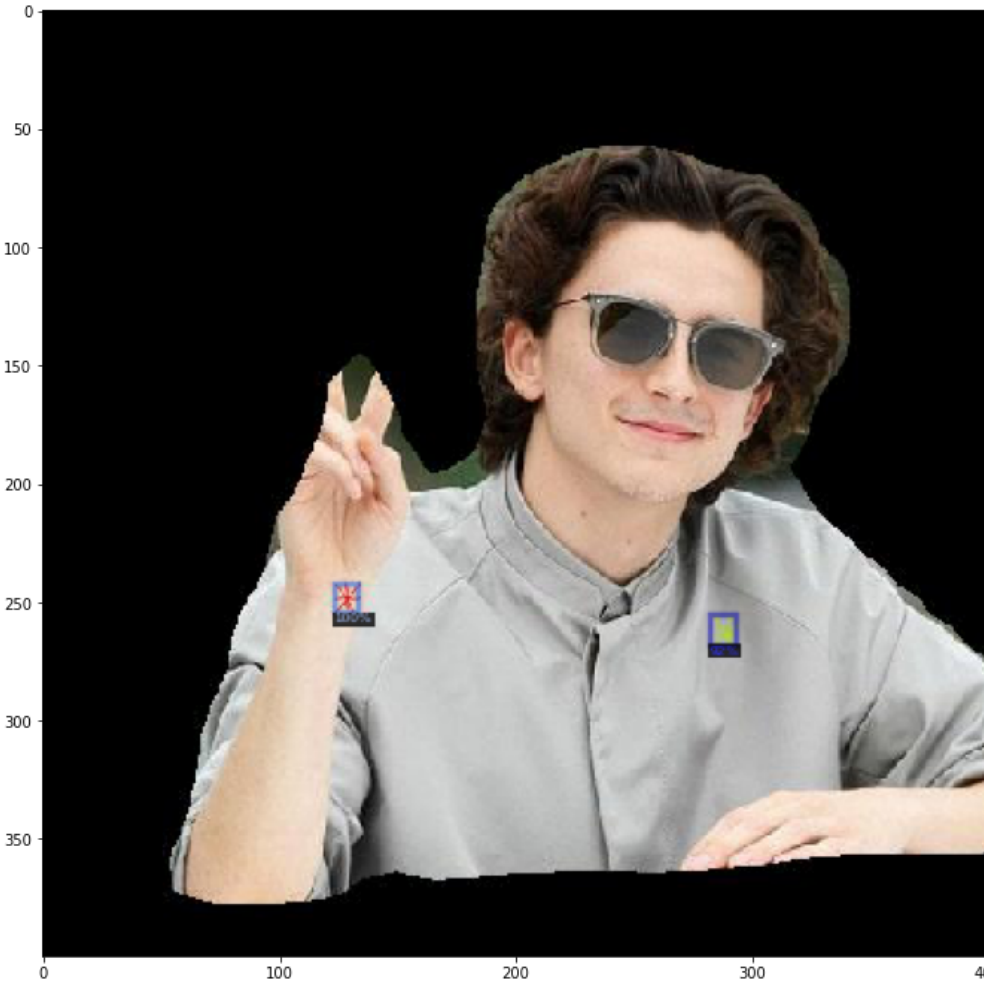
Step #2 - Detect Marks

In [8]:
```
import torch, torchvision
import detectron2
from detectron2.utils.logger import setup_logger
setup_logger()
from detectron2.engine import DefaultPredictor
from detectron2.utils.visualizer import Visualizer, ColorMo
from detectron2.evaluation import COCOEvaluator, inference_
from detectron2.data import MetadataCatalog
```

In [10]:
```
OUTPUT_DIR = "/media/farhat/Research/GitHub/Mark-NET/output
filename = OUTPUT_DIR + "2020-10-03 23_31 afterAll_15k_iter

import pickle
with open(filename, 'rb') as f:
     cfg = pickle.load(f)

# --- Evaluation Setup
cfg.MODEL.WEIGHTS = os.path.join(cfg.OUTPUT_DIR, "model_fin
cfg.MODEL.ROI_HEADS.SCORE_THRESH_TEST = 0.85

predictor = DefaultPredictor(cfg)
```

In [15]:
```
imgs = [skimage.io.imread(out_dir + path) for path in os.li
names = [path for path in os.listdir(out_dir) if ".jpg" in

import matplotlib.pyplot as plt

plt.figure(figsize=(30,30))
columns = 2
if not os.path.exists(out_dir + '/marks'): os.makedirs(out_
else:
    shutil.rmtree(out_dir + '/marks')
    os.makedirs(out_dir + '/marks')

for i, img in enumerate(imgs):
    outputs = predictor(img)
    v = Visualizer(img[:, :, ::-1],
       MetadataCatalog.get(cfg.DATASETS.TRAIN[0]),
       scale=0.8,
       instance_mode=ColorMode.IMAGE_BW)
    v = v.draw_instance_predictions(outputs["instances"].to
    img = v.get_image()[:, :, ::-1]
    plt.subplot(len(ids) / columns + 1, columns, i + 1)
    plt.imshow(img)
    skimage.io.imsave(f'{out_dir}/marks/{names[i]}', img)
```

```
/home/farhat/Farhat_files/Envs/mark_c3.6/lib/python3.6/site
pykernel_launcher.py:21: MatplotlibDeprecationWarning: Pass
egers as three-element position specification is deprecated
and will be removed two minor releases later.
```



In [ ]:

In [ ]:

In [ ]:

In [ ]: