In [1]:
```
import os
import sys
import time
import numpy as np
import imgaug
from pycocotools.coco import COCO
from pycocotools.cocoeval import COCOeval
from pycocotools import mask as maskUtils
import zipfile
import urllib.request
import json
import shutil
from PIL import Image, ImageDraw
```

In [2]:
```
# Root directory of the project
ROOT_DIR = os.path.abspath("../../../Mask_RCNN/")
# Import Mask RCNN
# Import mrcnn libraries
sys.path.append(ROOT_DIR)
from mrcnn.config import Config
import mrcnn.utils as utils
from mrcnn import visualize
import mrcnn.model as modellib
```

```
Using TensorFlow backend.
```

In [27]:
```
# Directory to save logs and trained model
MODEL_DIR = os.path.join(ROOT_DIR, "logs")

# Local path to trained weights file
COCO_MODEL_PATH = os.path.join(ROOT_DIR, "mask_rcnn_coco.h5")

# Download COCO trained weights from Releases if needed
if not os.path.exists(COCO_MODEL_PATH):
    utils.download_trained_weights(COCO_MODEL_PATH)
DEFAULT_DATASET_YEAR = "2018"
```

In [4]:
```
dir = '../../../skin/'
```

In [5]:
```
############################################################
#  Configurations
############################################################

class SkinConfig(Config):
    """Configuration for training on the toy  dataset.
    Derives from the base Config class and overrides some values.
    """
    # Give the configuration a recognizable name
    NAME = "skin"

    # We use a GPU with 12GB memory, which can fit two images.
    # Adjust down if you use a smaller GPU.
    IMAGES_PER_GPU = 2

    # Number of classes (including background)
    NUM_CLASSES = 1 + 1  # Background + skin

    # Number of training steps per epoch
    STEPS_PER_EPOCH = 100

    # This is how often validation is run. If you are using too much hard d
rive space
    # on saved models (in the MODEL_DIR), try making this value larger.
    VALIDATION_STEPS = 5

    # Matterport originally used resnet101, but I downsized to fit it on my
graphics card
    BACKBONE = 'resnet50'

    # Skip detections with < 90% confidence
#     DETECTION_MIN_CONFIDENCE = 0.9
     # To be honest, I haven't taken the time to figure out what these do
    RPN_ANCHOR_SCALES = (8, 16, 32, 64, 128)
    TRAIN_ROIS_PER_IMAGE = 32
    MAX_GT_INSTANCES = 50
    POST_NMS_ROIS_INFERENCE = 500
    POST_NMS_ROIS_TRAINING = 1000


config = SkinConfig()
config.display()
```

```
Configurations:
BACKBONE                       resnet50
BACKBONE_STRIDES               [4, 8, 16, 32, 64]
BATCH_SIZE                     2
BBOX_STD_DEV                   [0.1 0.1 0.2 0.2]
COMPUTE_BACKBONE_SHAPE         None
DETECTION_MAX_INSTANCES        100
DETECTION_MIN_CONFIDENCE       0.7
DETECTION_NMS_THRESHOLD        0.3
FPN_CLASSIF_FC_LAYERS_SIZE     1024
GPU_COUNT                      1
GRADIENT_CLIP_NORM             5.0
IMAGES_PER_GPU                 2
IMAGE_CHANNEL_COUNT            3
IMAGE_MAX_DIM                  1024
IMAGE_META_SIZE                14
IMAGE_MIN_DIM                  800
IMAGE_MIN_SCALE                0
IMAGE_RESIZE_MODE              square
IMAGE_SHAPE                    [1024 1024    3]
LEARNING_MOMENTUM              0.9
LEARNING_RATE                  0.001
LOSS_WEIGHTS                   {'rpn_bbox_loss': 1.0, 'mrcnn_mask_loss': 1.
0, 'rpn_class_loss': 1.0, 'mrcnn_class_loss': 1.0, 'mrcnn_bbox_loss': 1.0}
MASK_POOL_SIZE                 14
MASK_SHAPE                     [28, 28]
MAX_GT_INSTANCES               50
MEAN_PIXEL                     [123.7 116.8 103.9]
MINI_MASK_SHAPE                (56, 56)
NAME                           skin
NUM_CLASSES                    2
POOL_SIZE                      7
POST_NMS_ROIS_INFERENCE        500
POST_NMS_ROIS_TRAINING         1000
PRE_NMS_LIMIT                  6000
ROI_POSITIVE_RATIO             0.33
RPN_ANCHOR_RATIOS              [0.5, 1, 2]
RPN_ANCHOR_SCALES              (8, 16, 32, 64, 128)
RPN_ANCHOR_STRIDE              1
RPN_BBOX_STD_DEV               [0.1 0.1 0.2 0.2]
RPN_NMS_THRESHOLD              0.7
RPN_TRAIN_ANCHORS_PER_IMAGE    256
STEPS_PER_EPOCH                100
TOP_DOWN_PYRAMID_SIZE          256
TRAIN_BN                       False
TRAIN_ROIS_PER_IMAGE           32
USE_MINI_MASK                  True
USE_RPN_ROIS                   True
VALIDATION_STEPS               5
WEIGHT_DECAY                   0.0001
```

In [6]:
```
c = SkinDataset()
c.load_data('../../../skin/train/instances_skin_train2019.json','../../../s
kin/train/skin_train2019/')
```

```
---------------------------------------
NameErrorTraceback (most recent call last)
<ipython-input-6-f307405a2523> in <module>
----> 1 c = SkinDataset()
      2 c.load_data('../../../skin/train/instances_skin_train2019.json','..
/../../skin/train/skin_train2019/')

NameError: name 'SkinDataset' is not defined
```

In [14]:
```
class SkinDataset(utils.Dataset):

    def load_data(self, annotation_json, images_dir):
        """ Load the coco-like dataset from json
        Args:
            annotation_json: The path to the coco annotations json file
            images_dir: The directory holding the images referred to by the
json file
        """
        # Load json from file
        json_file = open(annotation_json)
        coco_json = json.load(json_file)
        json_file.close()

        # Add the class names using the base method from utils.Dataset
        source_name = "skin"
        for category in coco_json['categories']:
            class_id = category['id']
            class_name = category['name']
#             print(class_name)

            if class_id < 1:
                print('Error: Class id for "{}" cannot be less than one. (0
is reserved for the background)'.format(class_name))
                return

            self.add_class(source_name, class_id, class_name)
#             print('hi')

        # Get all annotations
        annotations = {}
        for annotation in coco_json['annotations']:
            image_id = annotation['image_id']
#             print(image_id)

            if image_id not in annotations:
                annotations[image_id] = []
            annotations[image_id].append(annotation)
#             print(annotations)


        # Get all images and add them to the dataset
        seen_images = {}
        for image in coco_json['images']:
            image_id = image['id']
            if image_id in seen_images:
                print("Warning: Skipping duplicate image id: {}".format(ima
ge))
            else:
                seen_images[image_id] = image
                try:
                    image_file_name = image['file_name']
                    image_width = image['width']
                    image_height = image['height']
                except KeyError as key:
                    print("Warning: Skipping image (id: {}) with missing ke
y: {}".format(image_id, key))

                image_path = os.path.abspath(os.path.join(images_dir, image
_file_name))
                image_annotations = annotations[image_id]

                # Add the image using the base method from utils.Dataset
```

Creating Training and Validation Data

In [15]:
```
dataset_train = SkinDataset()
dataset_train.load_data('../../../skin/train/instances_skin_train2019.json'
,'../../../skin/train/skin_train2019/')
dataset_train.prepare()
```

In [24]:
```
dataset_val = SkinDataset()
dataset_val.load_data('../../../skin/val/instances_skin_val2019.json','../.
./../skin/val/skin_val2019/')
dataset_val.prepare()
```

Display a few images from the training dataset¶

In [25]:
```
dataset = dataset_train
image_ids = np.random.choice(dataset.image_ids, 4)

for image_id in image_ids:
    image = dataset.load_image(image_id)
    mask, class_ids = dataset.load_mask(image_id)
    visualize.display_top_masks(image, mask, class_ids, dataset.class_names
)
```

Create the Training Model and Train¶

In [28]:
```
# Create model in training mode
model = modellib.MaskRCNN(mode="training", config=config,
                          model_dir=MODEL_DIR)
```

In [29]:
```
# Which weights to start with?
init_with = "coco"  # imagenet, coco, or last

if init_with == "imagenet":
    model.load_weights(model.get_imagenet_weights(), by_name=True)
elif init_with == "coco":
    # Load weights trained on MS COCO, but skip layers that
    # are different due to the different number of classes
    # See README for instructions to download the COCO weights
    model.load_weights(COCO_MODEL_PATH, by_name=True,
                       exclude=["mrcnn_class_logits", "mrcnn_bbox_fc",
                                "mrcnn_bbox", "mrcnn_mask"])
elif init_with == "last":
    # Load the last model you trained and continue training
    model.load_weights(model.find_last(), by_name=True)
```

Training

Train in two stages:

- Only the heads. Here we're freezing all the backbone layers and training only the randomly initialized layers (i.e. the ones that we didn't use pre-trained weights from MS COCO). To train only the head layers, pass layers='heads' to the train() function.
- Fine-tune all layers. For this simple example it's not necessary, but we're including it to show the process. Simply pass layers="all to train all layers.

In [33]:
```
# Train the head branches
# Passing layers="heads" freezes all layers except the head
# layers. You can also pass a regular expression to select
# which layers to train by name pattern.
start_train = time.time()
model.train(dataset_train, dataset_val,
            learning_rate=config.LEARNING_RATE,
            epochs=4,
            layers='heads')
end_train = time.time()
minutes = round((end_train - start_train) / 60, 2)
print('Training took' + str(minutes)+ 'minutes')
```

```
Starting at epoch 0. LR=0.001

Checkpoint Path: /home/anirudh/detect/code/Umbilicus_Skin_Detection/Mask_RC
NN/logs/skin20190212T0154/mask_rcnn_skin_{epoch:04d}.h5
Selecting layers to train
fpn_c5p5               (Conv2D)
fpn_c4p4               (Conv2D)
fpn_c3p3               (Conv2D)
fpn_c2p2               (Conv2D)
fpn_p5                 (Conv2D)
fpn_p2                 (Conv2D)
fpn_p3                 (Conv2D)
fpn_p4                 (Conv2D)
In model:  rpn_model
    rpn_conv_shared        (Conv2D)
    rpn_class_raw          (Conv2D)
    rpn_bbox_pred          (Conv2D)
mrcnn_mask_conv1       (TimeDistributed)
mrcnn_mask_bn1         (TimeDistributed)
mrcnn_mask_conv2       (TimeDistributed)
mrcnn_mask_bn2         (TimeDistributed)
mrcnn_class_conv1      (TimeDistributed)
mrcnn_class_bn1        (TimeDistributed)
mrcnn_mask_conv3       (TimeDistributed)
mrcnn_mask_bn3         (TimeDistributed)
mrcnn_class_conv2      (TimeDistributed)
mrcnn_class_bn2        (TimeDistributed)
mrcnn_mask_conv4       (TimeDistributed)
mrcnn_mask_bn4         (TimeDistributed)
mrcnn_bbox_fc          (TimeDistributed)
mrcnn_mask_deconv      (TimeDistributed)
mrcnn_class_logits     (TimeDistributed)
mrcnn_mask             (TimeDistributed)

/home/anirudh/detect/lib/python3.5/site-packages/tensorflow/python/ops/grad
ients_impl.py:108: UserWarning: Converting sparse IndexedSlices to a dense
Tensor of unknown shape. This may consume a large amount of memory.
  "Converting sparse IndexedSlices to a dense Tensor of unknown shape. "
/home/anirudh/detect/lib/python3.5/site-packages/keras/engine/training_gene
rator.py:47: UserWarning: Using a generator with `use_multiprocessing=True`
and multiple workers may duplicate your data. Please consider using the`ker
as.utils.Sequence class.
  UserWarning('Using a generator with `use_multiprocessing=True`'
```

```
Epoch 1/4
100/100 [==============================] - 749s 7s/step - loss: 4.0688 - rp
n_class_loss: 0.0692 - rpn_bbox_loss: 3.1221 - mrcnn_class_loss: 0.0964 - m
rcnn_bbox_loss: 0.3515 - mrcnn_mask_loss: 0.4296 - val_loss: 2.3637 - val_r
pn_class_loss: 0.0317 - val_rpn_bbox_loss: 1.8365 - val_mrcnn_class_loss: 0
.0417 - val_mrcnn_bbox_loss: 0.1832 - val_mrcnn_mask_loss: 0.2706
Epoch 2/4
100/100 [==============================] - 715s 7s/step - loss: 1.7120 - rp
n_class_loss: 0.0302 - rpn_bbox_loss: 1.1581 - mrcnn_class_loss: 0.0674 - m
rcnn_bbox_loss: 0.2279 - mrcnn_mask_loss: 0.2285 - val_loss: 2.6643 - val_r
pn_class_loss: 0.0194 - val_rpn_bbox_loss: 2.2787 - val_mrcnn_class_loss: 0
.0661 - val_mrcnn_bbox_loss: 0.1298 - val_mrcnn_mask_loss: 0.1703
Epoch 3/4
100/100 [==============================] - 710s 7s/step - loss: 2.3434 - rp
n_class_loss: 0.0516 - rpn_bbox_loss: 1.7491 - mrcnn_class_loss: 0.0979 - m
rcnn_bbox_loss: 0.2253 - mrcnn_mask_loss: 0.2195 - val_loss: 2.7694 - val_r
pn_class_loss: 0.0226 - val_rpn_bbox_loss: 2.2331 - val_mrcnn_class_loss: 0
.0880 - val_mrcnn_bbox_loss: 0.2772 - val_mrcnn_mask_loss: 0.1484
Epoch 4/4
100/100 [==============================] - 695s 7s/step - loss: 2.7218 - rp
n_class_loss: 0.0474 - rpn_bbox_loss: 2.2012 - mrcnn_class_loss: 0.1069 - m
rcnn_bbox_loss: 0.1867 - mrcnn_mask_loss: 0.1796 - val_loss: 4.1684 - val_r
pn_class_loss: 0.0193 - val_rpn_bbox_loss: 3.4843 - val_mrcnn_class_loss: 0
.0869 - val_mrcnn_bbox_loss: 0.4187 - val_mrcnn_mask_loss: 0.1593
Training took48.22minutes
```

In [34]:
```
# Fine tune all layers
# Passing layers="all" trains all layers. You can also
# pass a regular expression to select which layers to
# train by name pattern.
start_train = time.time()
model.train(dataset_train, dataset_val,
            learning_rate=config.LEARNING_RATE / 10,
            epochs=8,
            layers="all")
end_train = time.time()
minutes = round((end_train - start_train) / 60, 2)
print('Training took' + str(minutes)+ 'minutes')
```

```
Starting at epoch 4. LR=0.0001

Checkpoint Path: /home/anirudh/detect/code/Umbilicus_Skin_Detection/Mask_RC
NN/logs/skin20190212T0154/mask_rcnn_skin_{epoch:04d}.h5
Selecting layers to train
conv1                  (Conv2D)
bn_conv1               (BatchNorm)
res2a_branch2a         (Conv2D)
bn2a_branch2a          (BatchNorm)
res2a_branch2b         (Conv2D)
bn2a_branch2b          (BatchNorm)
res2a_branch2c         (Conv2D)
res2a_branch1          (Conv2D)
bn2a_branch2c          (BatchNorm)
bn2a_branch1           (BatchNorm)
res2b_branch2a         (Conv2D)
bn2b_branch2a          (BatchNorm)
res2b_branch2b         (Conv2D)
bn2b_branch2b          (BatchNorm)
res2b_branch2c         (Conv2D)
bn2b_branch2c          (BatchNorm)
res2c_branch2a         (Conv2D)
bn2c_branch2a          (BatchNorm)
res2c_branch2b         (Conv2D)
bn2c_branch2b          (BatchNorm)
res2c_branch2c         (Conv2D)
bn2c_branch2c          (BatchNorm)
res3a_branch2a         (Conv2D)
bn3a_branch2a          (BatchNorm)
res3a_branch2b         (Conv2D)
bn3a_branch2b          (BatchNorm)
res3a_branch2c         (Conv2D)
res3a_branch1          (Conv2D)
bn3a_branch2c          (BatchNorm)
bn3a_branch1           (BatchNorm)
res3b_branch2a         (Conv2D)
bn3b_branch2a          (BatchNorm)
res3b_branch2b         (Conv2D)
bn3b_branch2b          (BatchNorm)
res3b_branch2c         (Conv2D)
bn3b_branch2c          (BatchNorm)
res3c_branch2a         (Conv2D)
bn3c_branch2a          (BatchNorm)
res3c_branch2b         (Conv2D)
bn3c_branch2b          (BatchNorm)
res3c_branch2c         (Conv2D)
bn3c_branch2c          (BatchNorm)
res3d_branch2a         (Conv2D)
bn3d_branch2a          (BatchNorm)
res3d_branch2b         (Conv2D)
bn3d_branch2b          (BatchNorm)
res3d_branch2c         (Conv2D)
bn3d_branch2c          (BatchNorm)
res4a_branch2a         (Conv2D)
bn4a_branch2a          (BatchNorm)
res4a_branch2b         (Conv2D)
bn4a_branch2b          (BatchNorm)
res4a_branch2c         (Conv2D)
res4a_branch1          (Conv2D)
bn4a_branch2c          (BatchNorm)
bn4a_branch1           (BatchNorm)
res4b_branch2a         (Conv2D)
bn4b_branch2a          (BatchNorm)
```

Prepare to run Inference

In [36]:
```
class InferenceConfig(SkinConfig):
    GPU_COUNT = 1
    IMAGES_PER_GPU = 1
    IMAGE_MIN_DIM = 512
    IMAGE_MAX_DIM = 512
    DETECTION_MIN_CONFIDENCE = 0.85


inference_config = InferenceConfig()
```

In [37]:
```
# Recreate the model in inference mode
model = modellib.MaskRCNN(mode="inference",
                          config=inference_config,
                          model_dir=MODEL_DIR)
```

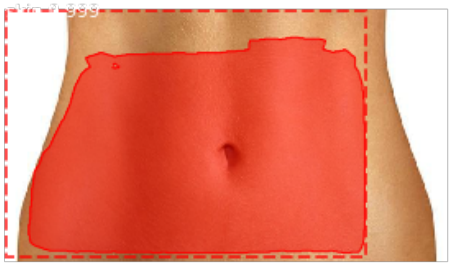
In [38]:
```
# Get path to saved weights
# Either set a specific path or find last trained weights
# model_path = os.path.join(ROOT_DIR, ".h5 file name here")
model_path = model.find_last()

# Load trained weights (fill in path to trained weights here)
assert model_path != "", "Provide path to trained weights"
print("Loading weights from ", model_path)
model.load_weights(model_path, by_name=True)
```

```
Loading weights from  /home/anirudh/detect/code/Umbilicus_Skin_Detection/Ma
sk_RCNN/logs/skin20190212T0154/mask_rcnn_skin_0008.h5
Re-starting from epoch 8
```

Run Inference

In [40]:
```
import skimage
real_test_dir = '../../../Input/user/testingdata/'
image_paths = []
for filename in os.listdir(real_test_dir):
    if os.path.splitext(filename)[1].lower() in ['.png', '.jpg', '.jpeg']:
        image_paths.append(os.path.join(real_test_dir, filename))

for image_path in image_paths:
    img = skimage.io.imread(image_path)
    img_arr = np.array(img)
    results = model.detect([img_arr], verbose=1)
    r = results[0]
    visualize.display_instances(img, r['rois'], r['masks'], r['class_ids'],
                                dataset_val.class_names, r['scores'], figsi
ze=(5,5))
```

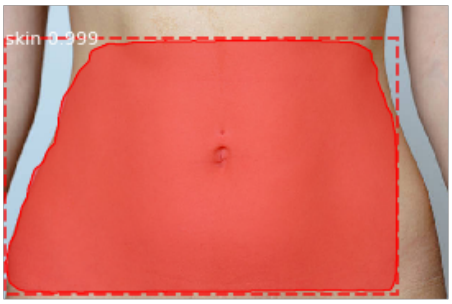
```
Processing 1 images
image                    shape: (340, 598, 3)         min:    0.00000  max:
255.00000  uint8
molded_images            shape: (1, 512, 512, 3)      min: -123.70000  max:
151.10000  float64
image_metas              shape: (1, 14)               min:    0.00000  max:
598.00000  float64
anchors                  shape: (1, 65472, 4)         min:   -0.17712  max:
1.05188  float32
```



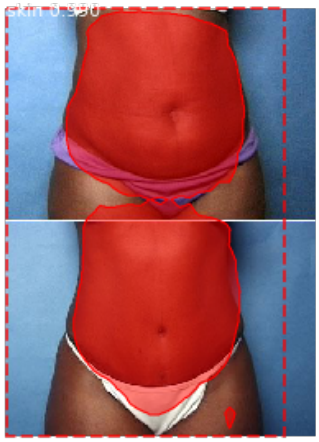
```
Processing 1 images
image                    shape: (682, 1023, 3)        min:    0.00000  max:
255.00000  uint8
molded_images            shape: (1, 512, 512, 3)      min: -123.70000  max:
151.10000  float64
image_metas              shape: (1, 14)               min:    0.00000  max:
1023.00000  float64
anchors                  shape: (1, 65472, 4)         min:   -0.17712  max:
1.05188  float32
```

```
Processing 1 images
image                    shape: (1870, 2420, 3)       min:    0.00000  max:
255.00000  uint8
molded_images            shape: (1, 512, 512, 3)      min: -123.70000  max:
151.10000  float64
image_metas              shape: (1, 14)               min:    0.00000  max:
2420.00000  float64
anchors                  shape: (1, 65472, 4)         min:   -0.17712  max:
1.05188  float32

*** No instances to display ***
```

```
Processing 1 images
image                    shape: (405, 612, 3)         min:    0.00000  max:
255.00000  uint8
molded_images            shape: (1, 512, 512, 3)      min: -123.70000  max:
143.10000  float64
image_metas              shape: (1, 14)               min:    0.00000  max:
612.00000  float64
anchors                  shape: (1, 65472, 4)         min:   -0.17712  max:
1.05188  float32
```




```
Processing 1 images
image                    shape: (478, 345, 3)         min:    0.00000  max:
255.00000  uint8
molded_images            shape: (1, 512, 512, 3)      min: -123.70000  max:
151.10000  float64
image_metas              shape: (1, 14)               min:    0.00000  max:
512.00000  float64
anchors                  shape: (1, 65472, 4)         min:   -0.17712  max:
1.05188  float32
```

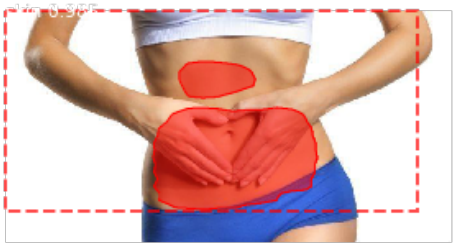


```
Processing 1 images
image                    shape: (355, 685, 3)         min:    0.00000  max:
255.00000  uint8
molded_images            shape: (1, 512, 512, 3)      min: -123.70000  max:
151.10000  float64
image_metas              shape: (1, 14)               min:    0.00000  max:
685.00000  float64
anchors                  shape: (1, 65472, 4)         min:   -0.17712  max:
1.05188  float32

*** No instances to display ***
```

```
Processing 1 images
image                    shape: (470, 300, 3)         min:    0.00000  max:
255.00000  uint8
molded_images            shape: (1, 512, 512, 3)      min: -123.70000  max:
151.10000  float64
image_metas              shape: (1, 14)               min:    0.00000  max:
512.00000  float64
anchors                  shape: (1, 65472, 4)         min:   -0.17712  max:
1.05188  float32
```

```
Processing 1 images
image                    shape: (300, 309, 3)         min:    0.00000  max:
255.00000  uint8
molded_images            shape: (1, 512, 512, 3)      min: -123.70000  max:
151.10000  float64
image_metas              shape: (1, 14)               min:    0.00000  max:
512.00000  float64
anchors                  shape: (1, 65472, 4)         min:   -0.17712  max:
1.05188  float32
```


```
Processing 1 images
image                    shape: (355, 685, 3)         min:    0.00000  max:
255.00000  uint8
molded_images            shape: (1, 512, 512, 3)      min: -123.70000  max:
151.10000  float64
image_metas              shape: (1, 14)               min:    0.00000  max:
685.00000  float64
anchors                  shape: (1, 65472, 4)         min:   -0.17712  max:
1.05188  float32
```

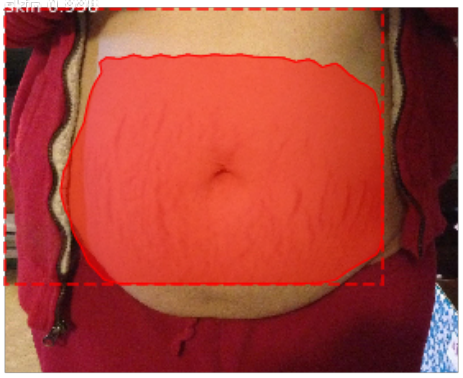


```
Processing 1 images
image                    shape: (377, 454, 3)         min:   38.00000  max:
255.00000  uint8
molded_images            shape: (1, 512, 512, 3)      min: -123.70000  max:
151.10000  float64
image_metas              shape: (1, 14)               min:    0.00000  max:
512.00000  float64
anchors                  shape: (1, 65472, 4)         min:   -0.17712  max:
1.05188  float32
```

```
Processing 1 images
image                    shape: (734, 1100, 3)        min:    0.00000  max:
255.00000  uint8
molded_images            shape: (1, 512, 512, 3)      min: -123.70000  max:
151.10000  float64
image_metas              shape: (1, 14)               min:    0.00000  max:
1100.00000  float64
anchors                  shape: (1, 65472, 4)         min:   -0.17712  max:
1.05188  float32
```

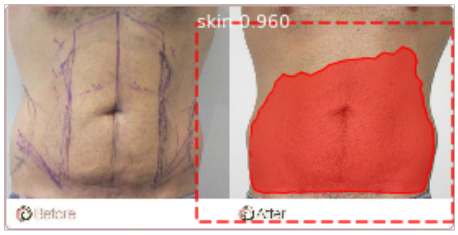
```
Processing 1 images
image                    shape: (250, 297, 3)         min:    0.00000  max:
255.00000  uint8
molded_images            shape: (1, 512, 512, 3)      min: -123.70000  max:
151.10000  float64
image_metas              shape: (1, 14)               min:    0.00000  max:
512.00000  float64
anchors                  shape: (1, 65472, 4)         min:   -0.17712  max:
1.05188  float32
```


```
Processing 1 images
image                    shape: (1290, 1600, 3)       min:    0.00000  max:
255.00000  uint8
molded_images            shape: (1, 512, 512, 3)      min: -123.70000  max:
151.10000  float64
image_metas              shape: (1, 14)               min:    0.00000  max:
1600.00000  float64
anchors                  shape: (1, 65472, 4)         min:   -0.17712  max:
1.05188  float32
```



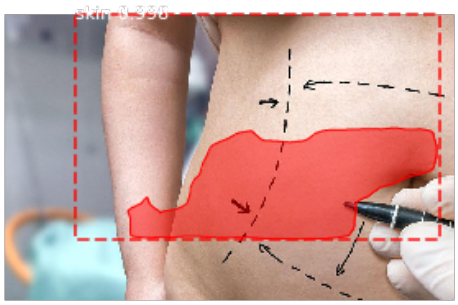
```
Processing 1 images
image                    shape: (355, 685, 3)         min:    0.00000  max:
255.00000  uint8
molded_images            shape: (1, 512, 512, 3)      min: -123.70000  max:
151.10000  float64
image_metas              shape: (1, 14)               min:    0.00000  max:
685.00000  float64
anchors                  shape: (1, 65472, 4)         min:   -0.17712  max:
1.05188  float32
```

```
Processing 1 images
image                    shape: (496, 992, 3)         min:    1.00000  max:
255.00000  uint8
molded_images            shape: (1, 512, 512, 3)      min: -123.70000  max:
151.10000  float64
image_metas              shape: (1, 14)               min:    0.00000  max:
992.00000  float64
anchors                  shape: (1, 65472, 4)         min:   -0.17712  max:
1.05188  float32
```

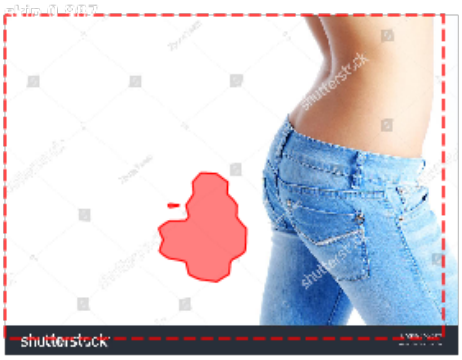



```
Processing 1 images
image                    shape: (408, 612, 3)         min:    0.00000  max:
255.00000  uint8
molded_images            shape: (1, 512, 512, 3)      min: -123.70000  max:
151.10000  float64
image_metas              shape: (1, 14)               min:    0.00000  max:
612.00000  float64
anchors                  shape: (1, 65472, 4)         min:   -0.17712  max:
1.05188  float32
```

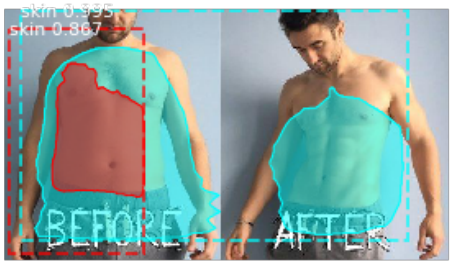
```
Processing 1 images
image                    shape: (636, 1000, 3)        min:    0.00000  max:
255.00000  uint8
molded_images            shape: (1, 512, 512, 3)      min: -123.70000  max:
151.10000  float64
image_metas              shape: (1, 14)               min:    0.00000  max:
1000.00000  float64
anchors                  shape: (1, 65472, 4)         min:   -0.17712  max:
1.05188  float32
```



```
Processing 1 images
image                    shape: (1126, 1500, 3)       min:    0.00000  max:
255.00000  uint8
molded_images            shape: (1, 512, 512, 3)      min: -123.70000  max:
151.10000  float64
image_metas              shape: (1, 14)               min:    0.00000  max:
1500.00000  float64
anchors                  shape: (1, 65472, 4)         min:   -0.17712  max:
1.05188  float32
```



```
Processing 1 images
image                    shape: (339, 602, 3)         min:    0.00000  max:
255.00000  uint8
molded_images            shape: (1, 512, 512, 3)      min: -123.70000  max:
151.10000  float64
image_metas              shape: (1, 14)               min:    0.00000  max:
602.00000  float64
anchors                  shape: (1, 65472, 4)         min:   -0.17712  max:
1.05188  float32
```




```
Processing 1 images
image                    shape: (265, 397, 3)         min:    3.00000  max:
255.00000  uint8
molded_images            shape: (1, 512, 512, 3)      min: -123.70000  max:
151.10000  float64
image_metas              shape: (1, 14)               min:    0.00000  max:
512.00000  float64
anchors                  shape: (1, 65472, 4)         min:   -0.17712  max:
1.05188  float32
```

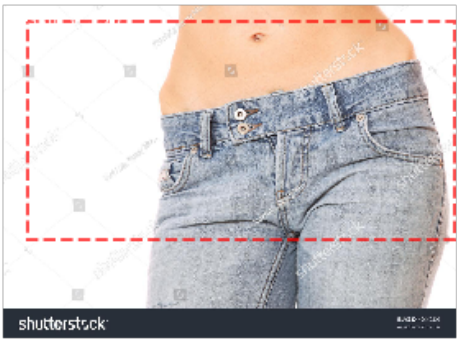

```
Processing 1 images
image                    shape: (786, 464, 3)         min:    0.00000  max:
255.00000  uint8
molded_images            shape: (1, 512, 512, 3)      min: -123.70000  max:
151.10000  float64
image_metas              shape: (1, 14)               min:    0.00000  max:
786.00000  float64
anchors                  shape: (1, 65472, 4)         min:   -0.17712  max:
1.05188  float32
```


```
Processing 1 images
image                    shape: (425, 450, 3)         min:    0.00000  max:
255.00000  uint8
molded_images            shape: (1, 512, 512, 3)      min: -123.70000  max:
151.10000  float64
image_metas              shape: (1, 14)               min:    0.00000  max:
512.00000  float64
anchors                  shape: (1, 65472, 4)         min:   -0.17712  max:
1.05188  float32

*** No instances to display ***
```


```
Processing 1 images
image                    shape: (469, 500, 3)         min:    0.00000  max:
255.00000  uint8
molded_images            shape: (1, 512, 512, 3)      min: -123.70000  max:
151.10000  float64
image_metas              shape: (1, 14)               min:    0.00000  max:
512.00000  float64
anchors                  shape: (1, 65472, 4)         min:   -0.17712  max:
1.05188  float32
```

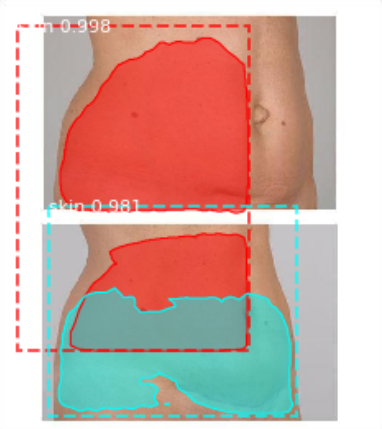

```
Processing 1 images
image                    shape: (383, 368, 3)         min:    0.00000  max:
255.00000  uint8
molded_images            shape: (1, 512, 512, 3)      min: -123.70000  max:
131.30000  float64
image_metas              shape: (1, 14)               min:    0.00000  max:
512.00000  float64
anchors                  shape: (1, 65472, 4)         min:   -0.17712  max:
1.05188  float32
```


```
Processing 1 images
image                    shape: (1101, 1500, 3)       min:    0.00000  max:
255.00000  uint8
molded_images            shape: (1, 512, 512, 3)      min: -123.70000  max:
151.10000  float64
image_metas              shape: (1, 14)               min:    0.00000  max:
1500.00000  float64
anchors                  shape: (1, 65472, 4)         min:   -0.17712  max:
1.05188  float32
```




```
Processing 1 images
image                    shape: (336, 702, 3)         min:    0.00000  max:
255.00000  uint8
molded_images            shape: (1, 512, 512, 3)      min: -123.70000  max:
151.10000  float64
image_metas              shape: (1, 14)               min:    0.00000  max:
702.00000  float64
anchors                  shape: (1, 65472, 4)         min:   -0.17712  max:
1.05188  float32

*** No instances to display ***
```

```
Processing 1 images
image                    shape: (581, 517, 3)         min:   19.00000  max:
255.00000  uint8
molded_images            shape: (1, 512, 512, 3)      min: -123.70000  max:
151.10000  float64
image_metas              shape: (1, 14)               min:    0.00000  max:
581.00000  float64
anchors                  shape: (1, 65472, 4)         min:   -0.17712  max:
1.05188  float32
```




```
Processing 1 images
image                    shape: (754, 482, 3)         min:    0.00000  max:
255.00000  uint8
molded_images            shape: (1, 512, 512, 3)      min: -123.70000  max:
117.30000  float64
image_metas              shape: (1, 14)               min:    0.00000  max:
754.00000  float64
anchors                  shape: (1, 65472, 4)         min:   -0.17712  max:
1.05188  float32
```


```
Processing 1 images
image                    shape: (1167, 1600, 3)       min:    0.00000  max:
255.00000  uint8
molded_images            shape: (1, 512, 512, 3)      min: -123.70000  max:
151.10000  float64
image_metas              shape: (1, 14)               min:    0.00000  max:
1600.00000  float64
anchors                  shape: (1, 65472, 4)         min:   -0.17712  max:
1.05188  float32
```

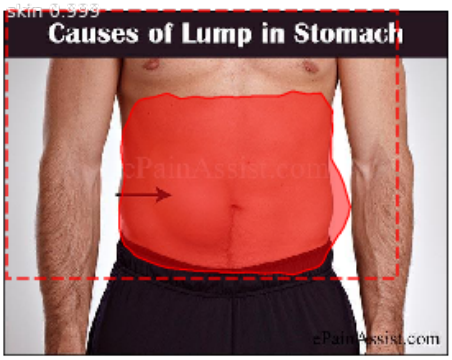
```
Processing 1 images
image                    shape: (336, 702, 3)         min:    0.00000  max:
255.00000  uint8
molded_images            shape: (1, 512, 512, 3)      min: -123.70000  max:
151.10000  float64
image_metas              shape: (1, 14)               min:    0.00000  max:
702.00000  float64
anchors                  shape: (1, 65472, 4)         min:   -0.17712  max:
1.05188  float32

*** No instances to display ***
```

```
Processing 1 images
image                    shape: (299, 400, 3)         min:    0.00000  max:
255.00000  uint8
molded_images            shape: (1, 512, 512, 3)      min: -123.70000  max:
142.10000  float64
image_metas              shape: (1, 14)               min:    0.00000  max:
512.00000  float64
anchors                  shape: (1, 65472, 4)         min:   -0.17712  max:
1.05188  float32
```

```
Processing 1 images
image                    shape: (405, 612, 3)         min:    0.00000  max:
255.00000  uint8
molded_images            shape: (1, 512, 512, 3)      min: -123.70000  max:
151.10000  float64
image_metas              shape: (1, 14)               min:    0.00000  max:
612.00000  float64
anchors                  shape: (1, 65472, 4)         min:   -0.17712  max:
1.05188  float32
```

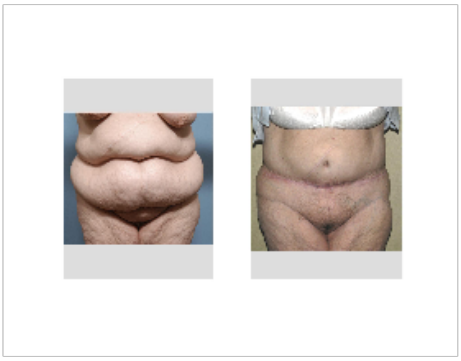
```
Processing 1 images
image                    shape: (1600, 1449, 3)       min:    0.00000  max:
255.00000  uint8
molded_images            shape: (1, 512, 512, 3)      min: -123.70000  max:
151.10000  float64
image_metas              shape: (1, 14)               min:    0.00000  max:
1600.00000  float64
anchors                  shape: (1, 65472, 4)         min:   -0.17712  max:
1.05188  float32
```


```
Processing 1 images
image                    shape: (516, 667, 3)         min:    0.00000  max:
255.00000  uint8
molded_images            shape: (1, 512, 512, 3)      min: -123.70000  max:
151.10000  float64
image_metas              shape: (1, 14)               min:    0.00000  max:
667.00000  float64
anchors                  shape: (1, 65472, 4)         min:   -0.17712  max:
1.05188  float32
```

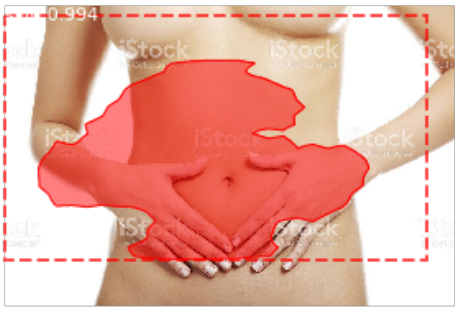



```
Processing 1 images
image                    shape: (660, 1022, 3)        min:    0.00000  max:
255.00000  uint8
molded_images            shape: (1, 512, 512, 3)      min: -123.70000  max:
151.10000  float64
image_metas              shape: (1, 14)               min:    0.00000  max:
1022.00000  float64
anchors                  shape: (1, 65472, 4)         min:   -0.17712  max:
1.05188  float32
```


```
Processing 1 images
image                    shape: (1870, 2420, 3)       min:    0.00000  max:
255.00000  uint8
molded_images            shape: (1, 512, 512, 3)      min: -123.70000  max:
151.10000  float64
image_metas              shape: (1, 14)               min:    0.00000  max:
2420.00000  float64
anchors                  shape: (1, 65472, 4)         min:   -0.17712  max:
1.05188  float32

*** No instances to display ***
```



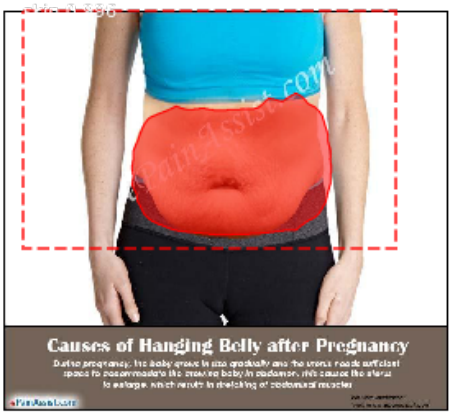
```
Processing 1 images
image                    shape: (683, 1024, 3)        min:    0.00000  max:
255.00000  uint8
molded_images            shape: (1, 512, 512, 3)      min: -123.70000  max:
151.10000  float64
image_metas              shape: (1, 14)               min:    0.00000  max:
1024.00000  float64
anchors                  shape: (1, 65472, 4)         min:   -0.17712  max:
1.05188  float32
```




```
Processing 1 images
image                    shape: (1390, 904, 3)        min:    0.00000  max:
255.00000  uint8
molded_images            shape: (1, 512, 512, 3)      min: -123.70000  max:
151.10000  float64
image_metas              shape: (1, 14)               min:    0.00000  max:
1390.00000  float64
anchors                  shape: (1, 65472, 4)         min:   -0.17712  max:
1.05188  float32
```


```
Processing 1 images
image                    shape: (470, 380, 3)         min:    0.00000  max:
255.00000  uint8
molded_images            shape: (1, 512, 512, 3)      min: -123.70000  max:
151.10000  float64
image_metas              shape: (1, 14)               min:    0.00000  max:
512.00000  float64
anchors                  shape: (1, 65472, 4)         min:   -0.17712  max:
1.05188  float32
```

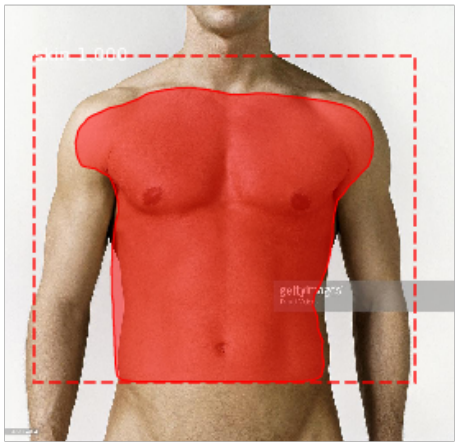

```
Processing 1 images
image                    shape: (600, 667, 3)         min:    0.00000  max:
255.00000  uint8
molded_images            shape: (1, 512, 512, 3)      min: -123.70000  max:
151.10000  float64
image_metas              shape: (1, 14)               min:    0.00000  max:
667.00000  float64
anchors                  shape: (1, 65472, 4)         min:   -0.17712  max:
1.05188  float32
```




```
Processing 1 images
image                    shape: (348, 618, 3)         min:    0.00000  max:
255.00000  uint8
molded_images            shape: (1, 512, 512, 3)      min: -123.70000  max:
151.10000  float64
image_metas              shape: (1, 14)               min:    0.00000  max:
618.00000  float64
anchors                  shape: (1, 65472, 4)         min:   -0.17712  max:
1.05188  float32
```

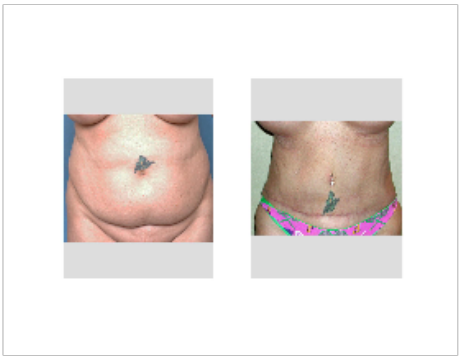
```
Processing 1 images
image                    shape: (1101, 1500, 3)       min:    0.00000  max:
255.00000  uint8
molded_images            shape: (1, 512, 512, 3)      min: -123.70000  max:
151.10000  float64
image_metas              shape: (1, 14)               min:    0.00000  max:
1500.00000  float64
anchors                  shape: (1, 65472, 4)         min:   -0.17712  max:
1.05188  float32
```


```
Processing 1 images
image                    shape: (994, 1024, 3)        min:    0.00000  max:
255.00000  uint8
molded_images            shape: (1, 512, 512, 3)      min: -123.70000  max:
148.10000  float64
image_metas              shape: (1, 14)               min:    0.00000  max:
1024.00000  float64
anchors                  shape: (1, 65472, 4)         min:   -0.17712  max:
1.05188  float32
```

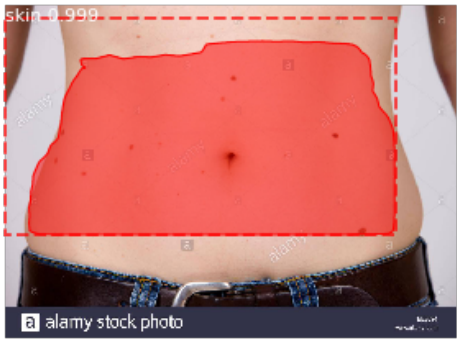


```
Processing 1 images
image                    shape: (2560, 2560, 3)       min:    0.00000  max:
255.00000  uint8
molded_images            shape: (1, 512, 512, 3)      min: -123.70000  max:
151.10000  float64
image_metas              shape: (1, 14)               min:    0.00000  max:
2560.00000  float64
anchors                  shape: (1, 65472, 4)         min:   -0.17712  max:
1.05188  float32

*** No instances to display ***
```

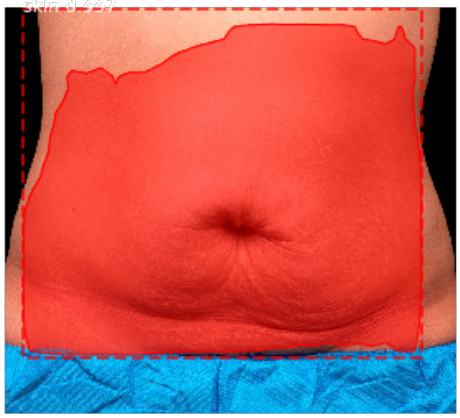
```
Processing 1 images
image                    shape: (1870, 2420, 3)       min:    0.00000  max:
255.00000  uint8
molded_images            shape: (1, 512, 512, 3)      min: -123.70000  max:
151.10000  float64
image_metas              shape: (1, 14)               min:    0.00000  max:
2420.00000  float64
anchors                  shape: (1, 65472, 4)         min:   -0.17712  max:
1.05188  float32

*** No instances to display ***
```



```
Processing 1 images
image                    shape: (956, 1300, 3)        min:    0.00000  max:
255.00000  uint8
molded_images            shape: (1, 512, 512, 3)      min: -123.70000  max:
151.10000  float64
image_metas              shape: (1, 14)               min:    0.00000  max:
1300.00000  float64
anchors                  shape: (1, 65472, 4)         min:   -0.17712  max:
1.05188  float32
```




```
Processing 1 images
image                    shape: (1084, 1200, 3)       min:    0.00000  max:
255.00000  uint8
molded_images            shape: (1, 512, 512, 3)      min: -123.70000  max:
151.10000  float64
image_metas              shape: (1, 14)               min:    0.00000  max:
1200.00000  float64
anchors                  shape: (1, 65472, 4)         min:   -0.17712  max:
1.05188  float32
```



```
Processing 1 images
image                    shape: (1024, 1021, 3)       min:    0.00000  max:
255.00000  uint8
molded_images            shape: (1, 512, 512, 3)      min: -123.70000  max:
151.10000  float64
image_metas              shape: (1, 14)               min:    0.00000  max:
1024.00000  float64
anchors                  shape: (1, 65472, 4)         min:   -0.17712  max:
1.05188  float32
```

In [ ]: