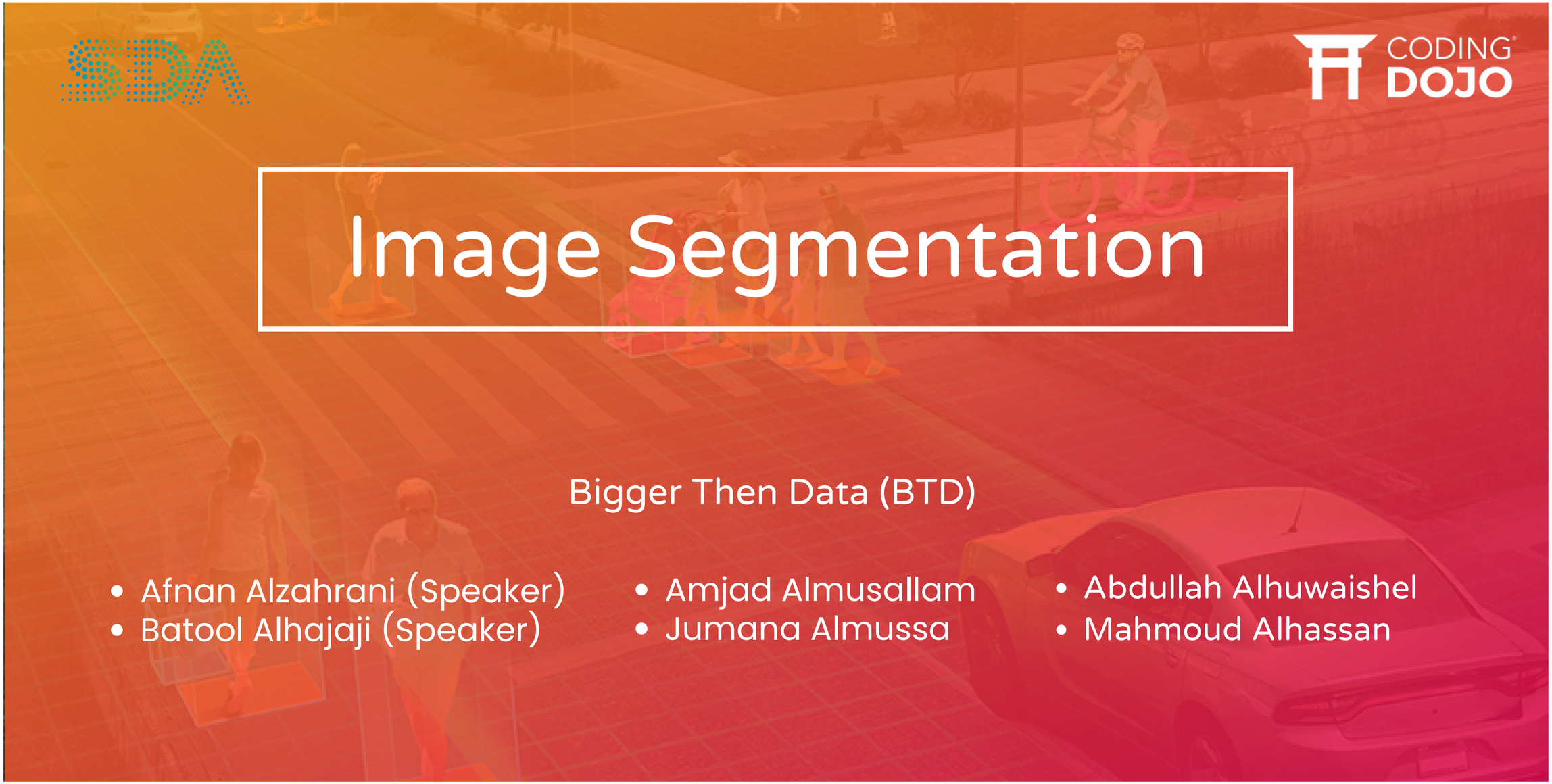
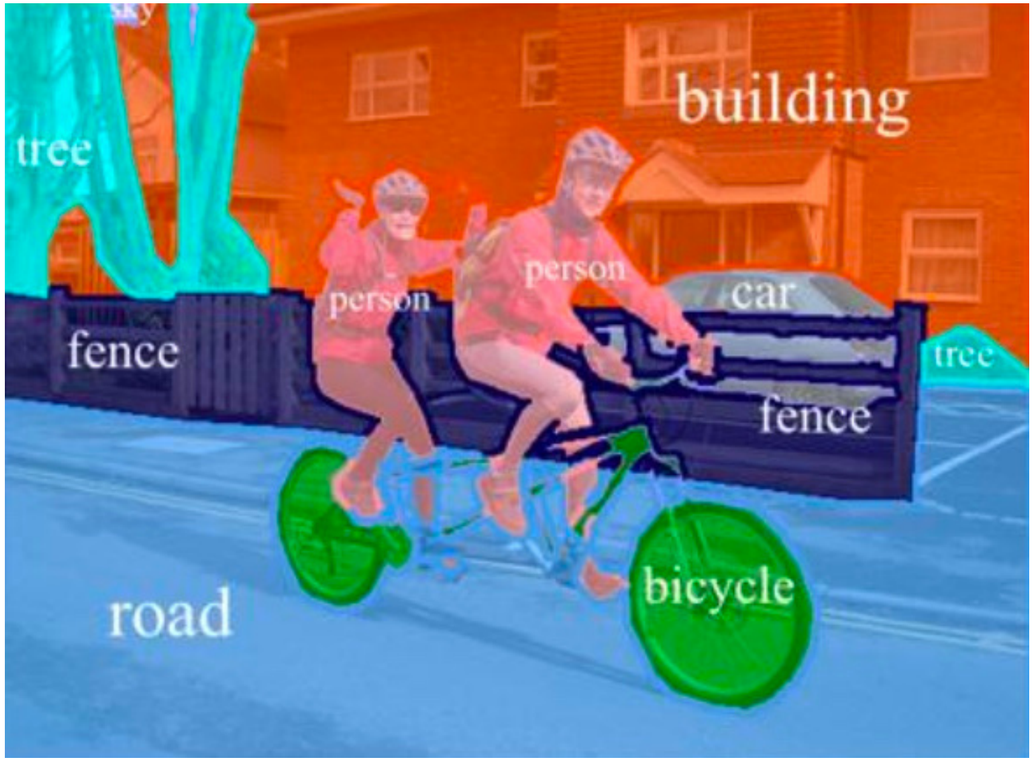


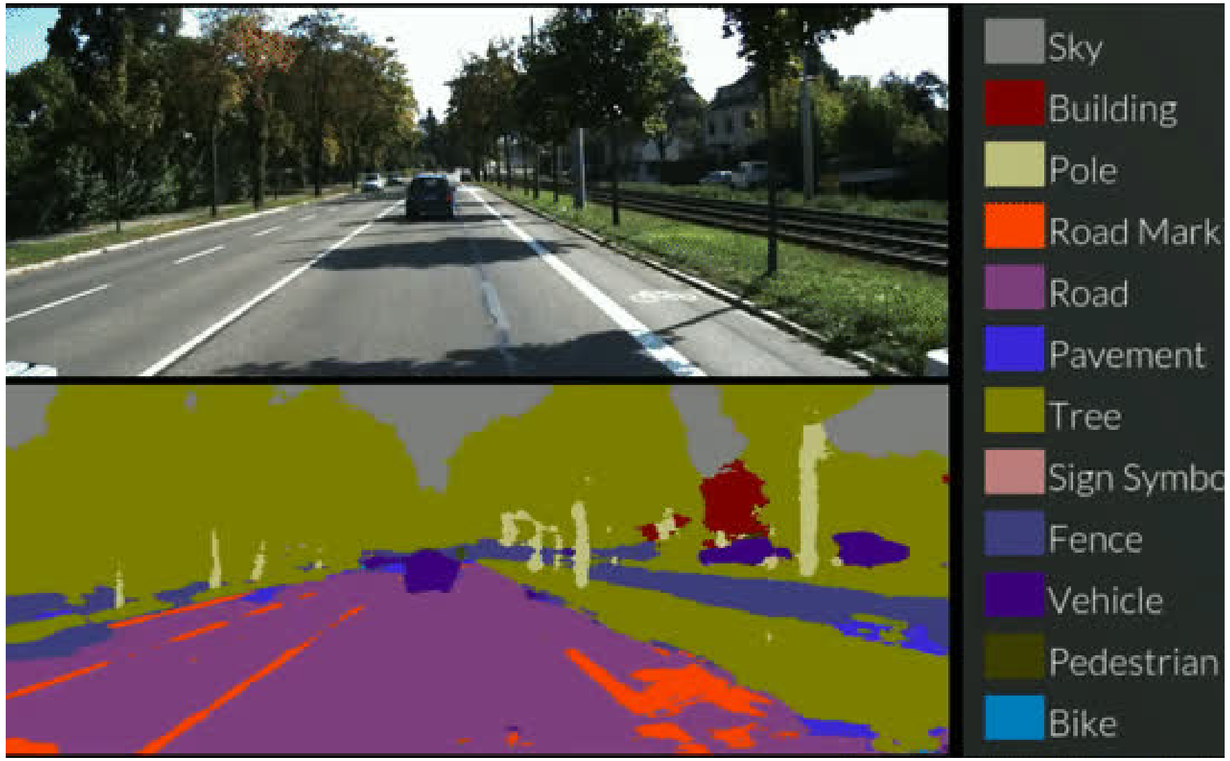
Image Segmentation

Bigger Then Data (BTD)

- Afnan Alzahrani (Speaker)
- Batool Alhajaji (Speaker)
- Amjad Almusallam
- Jumana Almussa
- Abdullah Alhuwaishel
- Mahmoud Alhassan

What is image segmentation?





One of the most important operations in Computer Vision

semantic image segmentation is the task of labelling each pixel of the image into a predefined set of classes.

Input

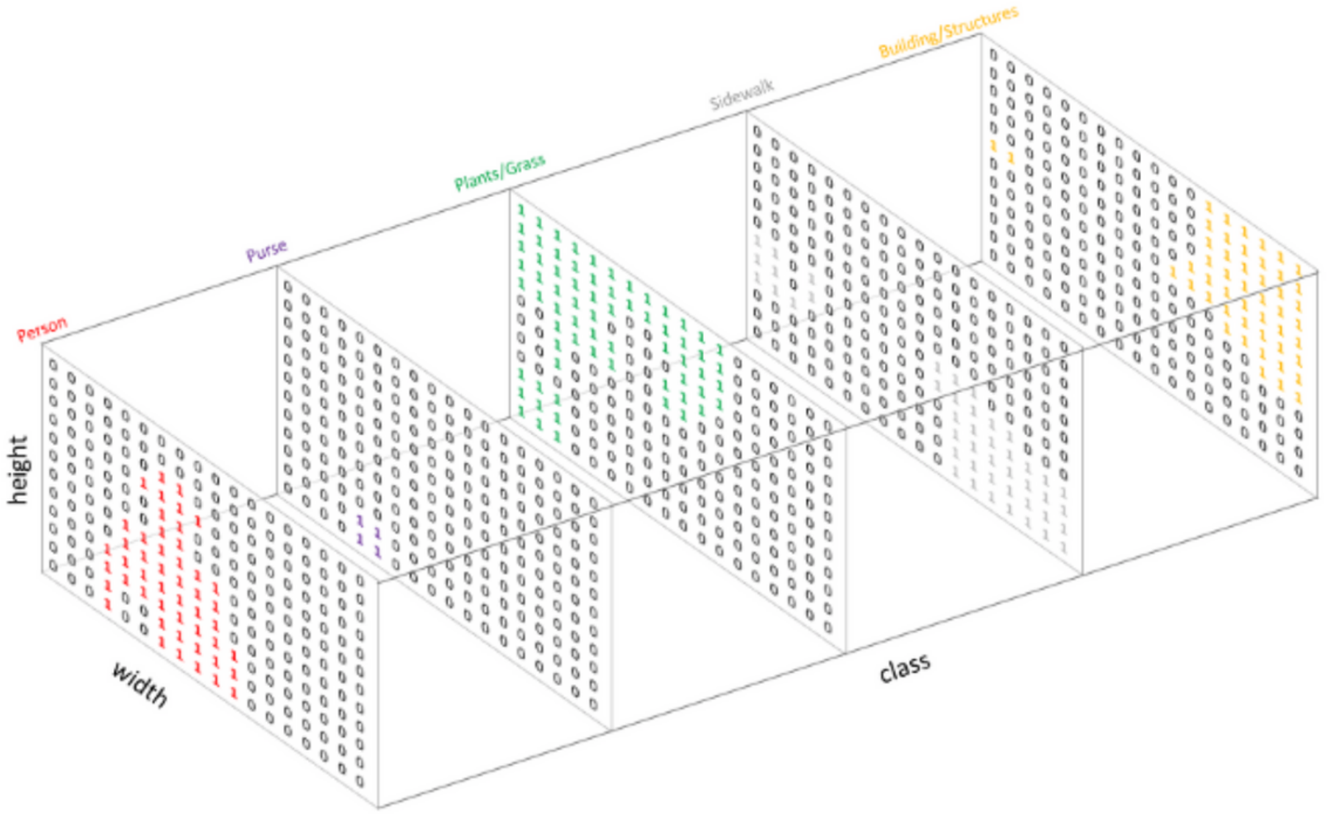
segmented

1: Person
2: Purse
3: Plants/Grass
4: Sidewalk
5: Building/Structures

3	3	3	3	3	3	3	3	3	3	3	3	5	5	5	5	5	5
3	3	3	3	3	3	3	3	3	3	3	3	5	5	5	5	5	5
3	3	3	3	3	3	1	1	3	3	3	3	5	5	5	5	5	5
3	3	3	3	3	1	1	1	1	3	3	3	5	5	5	5	5	5
3	3	3	3	3	3	1	1	3	3	3	5	5	5	5	5	5	5
5	5	3	3	3	3	1	1	3	3	5	5	5	5	5	5	5	5
4	4	3	4	1	1	1	1	1	1	4	4	4	5	5	5	5	5
4	4	3	4	1	1	1	1	1	1	4	4	4	4	4	5	5	5
4	4	4	1	1	1	1	1	1	1	1	4	4	4	4	4	4	4
3	3	3	1	1	1	1	1	1	1	1	4	4	4	4	4	4	4
3	3	3	1	2	2	1	1	1	1	1	4	4	4	4	4	4	4
3	3	3	1	2	2	1	1	1	1	1	4	4	4	4	4	4	4

Semantic Labels

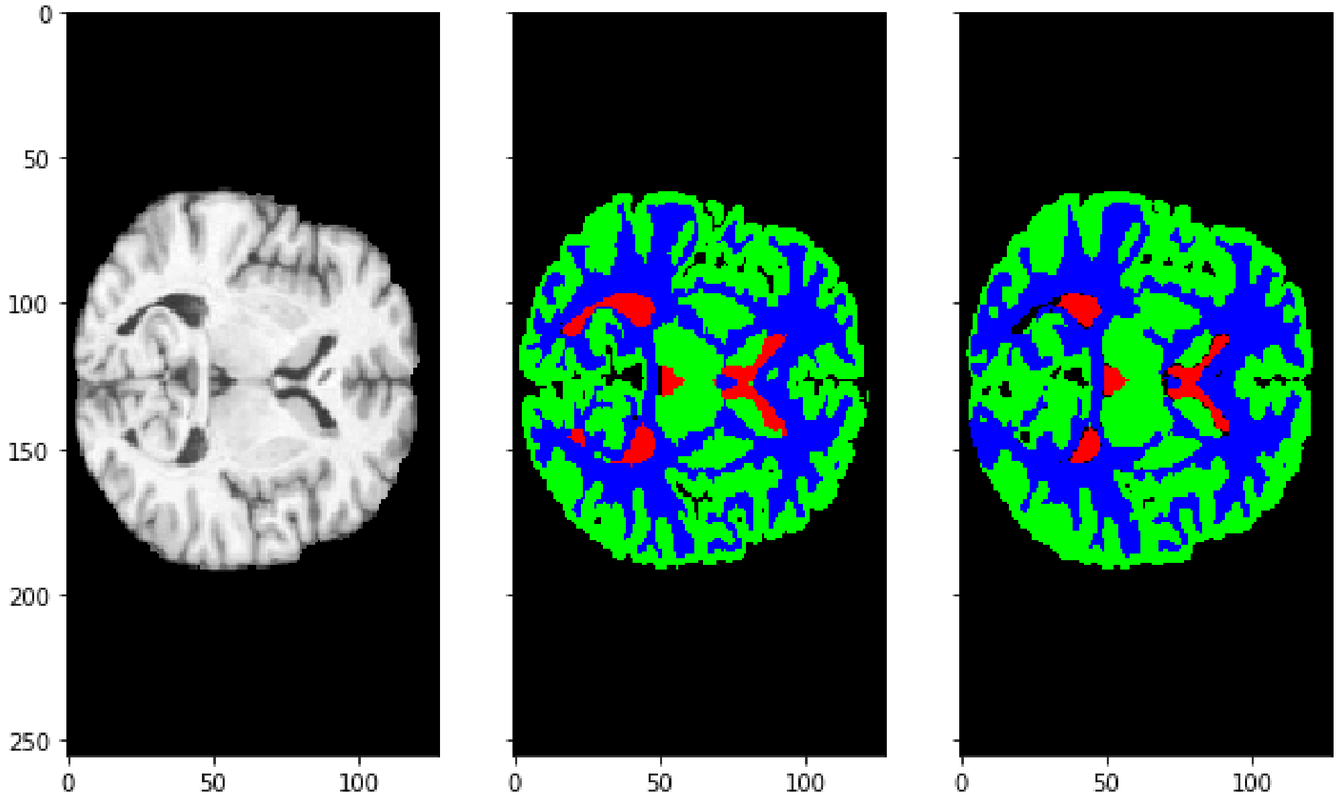
One-hot Encoded Segmentation Map



Applications

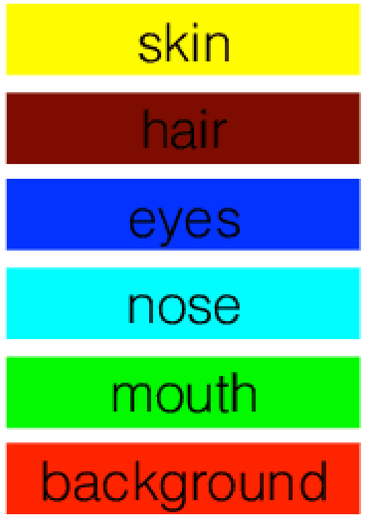
- Bio-medical image analysis

For example, models can be used to segment CT scans to detect tumours



Applications

- Facial segmentation



Computer vision using OpenCV

In [10]:
```
import matplotlib.pyplot as plt
```

In [11]:
```
import cv2
import numpy as np

with open('ENET/enet-classes.txt') as f:
    classes = f.read().splitlines()

with open('ENET/enet-colors.txt') as f:
    colors = f.read().splitlines()
    colors = [np.array(col.split(',')).astype('int') for col in colors]
    colors = np.array(colors,dtype='uint8')

model = cv2.dnn.readNet('ENET/enet-model.net')
cap = cv2.VideoCapture('telegram_video.mp4')
```

In [12]:
```
if not cap.isOpened():
    cap = cv2.VideoCapture(0)

if not cap.isOpened():
    raise IOError('Cannot Open Video File')
```

In [13]:
```
classes
```

Out[13]:
```
['Unlabeled',
 'Road',
 'Sidewalk',
 'Building',
 'Wall',
 'Fence',
 'Pole',
 'TrafficLight',
 'TrafficSign',
 'Vegetation',
 'Terrain',
 'Sky',
 'Person',
 'Rider',
 'Car',
 'Truck',
 'Bus',
 'Train',
 'Motorcycle',
 'Bicycle']
```

In [14]:
```
cap
```

Out[14]:
```
<VideoCapture 0x7f8e0694fbd0>
```

In [15]:
```
success,img = cap.read()
```

In [16]:
```
type(img)
```

Out[16]:
```
numpy.ndarray
```

In [17]:
```
plt.imshow(img)
```

Out[17]:
```
<matplotlib.image.AxesImage at 0x7f8e08d81250>
```

In [18]:
```
model
```

Out[18]:
```
<dnn_Net 0x7f8e0920ba10>
```

In [19]:
```
while True:
    success,img = cap.read()
    # Using size (1024,512) as ENET was trained on (1024,512) input size images.
    blob = cv2.dnn.blobFromImage(img,1/255,(1024,512),0,True,False)
    model.setInput(blob)
    output = model.forward()

    (numClasses, height, width) = output.shape[1:4]

    classMap = np.argmax(output[0], axis=0)
    mask = colors[classMap]
    mask = cv2.resize(mask, (img.shape[1], img.shape[0]),interpolation=cv2.INTER_NEAREST)
    output = ((0.3 * img) + (0.7 * mask)).astype("uint8")
    cv2.imshow('Output',output)
    cv2.imshow('Input',img)
    if cv2.waitKey(1) & 0xff==ord('q'):
        break
```

Thank You !

Source:
https://medium.com/analytics-vidhya/introduction-to-semantic-image-segmentation-856cda5e5de8