

In [254...

```
import cv2
import matplotlib.pyplot as plt
import numpy as np
import folium
from PIL import Image
import io
import math
plt.rcParams["figure.figsize"] = (50,5)
```

Nextbridge front road

Location: 31.475578921320253, 74.27099525928499

Google Map Url: [https://www.google.com/maps/place/Nextbridge+\(Pvt.\)+Ltd./@31.4745709,74.2711611,21z/data=!4m5!3m4!1s0x39190](https://www.google.com/maps/place/Nextbridge+(Pvt.)+Ltd./@31.4745709,74.2711611,21z/data=!4m5!3m4!1s0x39190)

In [256...

```
# next_lat,next_long = 31.475578921320253, 74.27099525928499
# duke_street_lat, duke_street_long = 51.51431455535977, -0.15399098396301272
lat, long = 31.475578921320253, 74.27099525928499
map_path = folium.Map(location=[lat, long],zoom_start=21,tiles='https://{s}.k
                    attr='&copy; <a href="https://www.openstreetmap.org/copyright
                    control_scale=False)
map_point = folium.Map(location=[lat, long],zoom_start=21,tiles='https://{s}.
                    attr='&copy; <a href="https://www.openstreetmap.org/copyright
                    control_scale=False)
folium.Circle(location=[lat, long],
              radius=1,
              fill=True,
              opacity=1,
              color="white"
              ).add_to(map_point)

img_data = map_path._to_png(5)
img = Image.open(io.BytesIO(img_data))
map_img = np.asarray(img)
map_img = map_img[0:map_img.shape[0]-20,45:]

img_data = map_point._to_png(5)
img = Image.open(io.BytesIO(img_data))
point_img = np.asarray(img)
point_img = point_img[0:point_img.shape[0]-20,45:]
```

1: Original Road Map

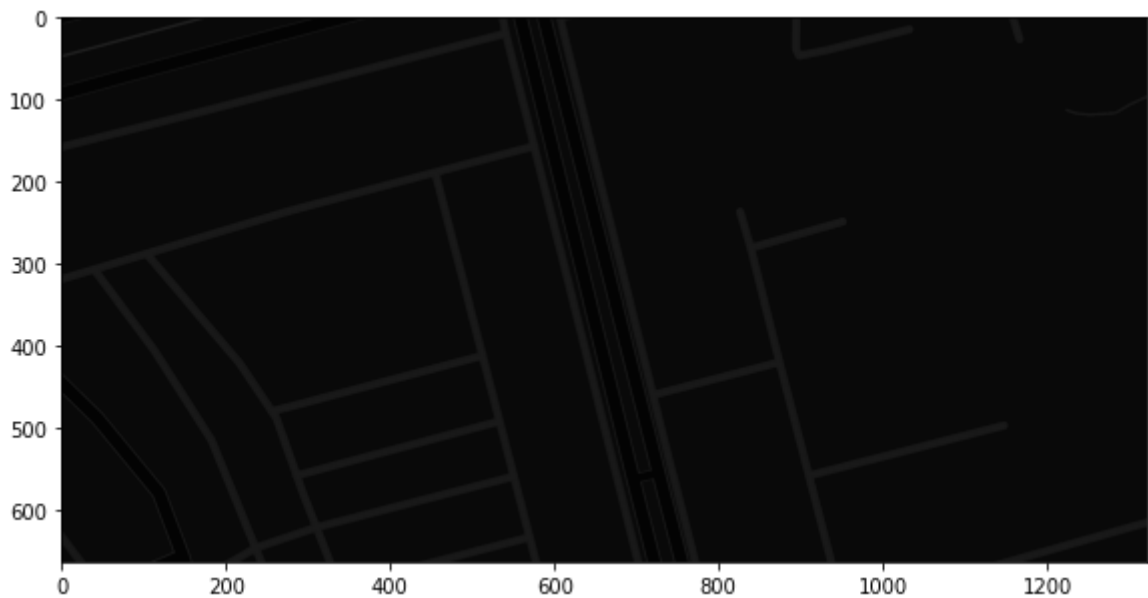
2: Point at which road is being measure

In [257...

```
plt.imshow(map_img)
```

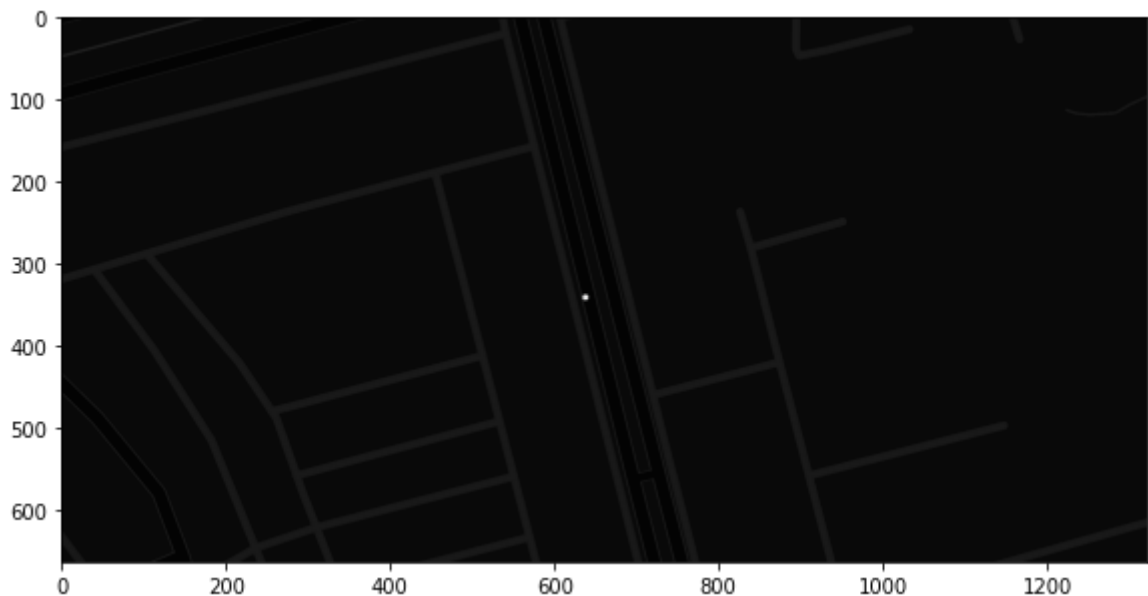
Out[257...

```
<matplotlib.image.AxesImage at 0x7fe5b6406df0>
```



In [258... `plt.imshow(point_img)`

Out[258... `<matplotlib.image.AxesImage at 0x7fe5b63a6cd0>`



In [259... `def road_segmentation(img):`
`img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)`
`img[img<=5]=0`
`img[(img>5)&(img<20)]=255`
`img[(img>=20)&(img<=30)]=50`
`return img`

`def get_point(img):`
`img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)`
`img[img>250]=255`
`img[img<255]=0`
`kernel = np.ones((5,5), np.uint8)`
`img = cv2.dilate(img, kernel, iterations=1)`
`contours, hierarchy = cv2.findContours(img, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)`
`for c in contours:`
`M = cv2.moments(c)`
`cX = int(M["m10"] / M["m00"])`
`cY = int(M["m01"] / M["m00"])`
`return cX, cY`

```
def get_two_points(cx,cy,img):
    val = img[cY][cX]
    points = []
    x1,y1 = cX,cY
    x2,y2 = cX,cY
    for e in [1,-1]:
        while True:
            if(e==1):
                x1 = x1+1
                new_val = img[y1][x1]
                if(val!=new_val):
                    points.append((x1,y1))
                    break
            else:
                continue
        else:
            x2 = x2-1
            new_val = img[y2][x2]
            if(val!=new_val):
                points.append((x2,y2))
                break
            else:
                continue
    return points

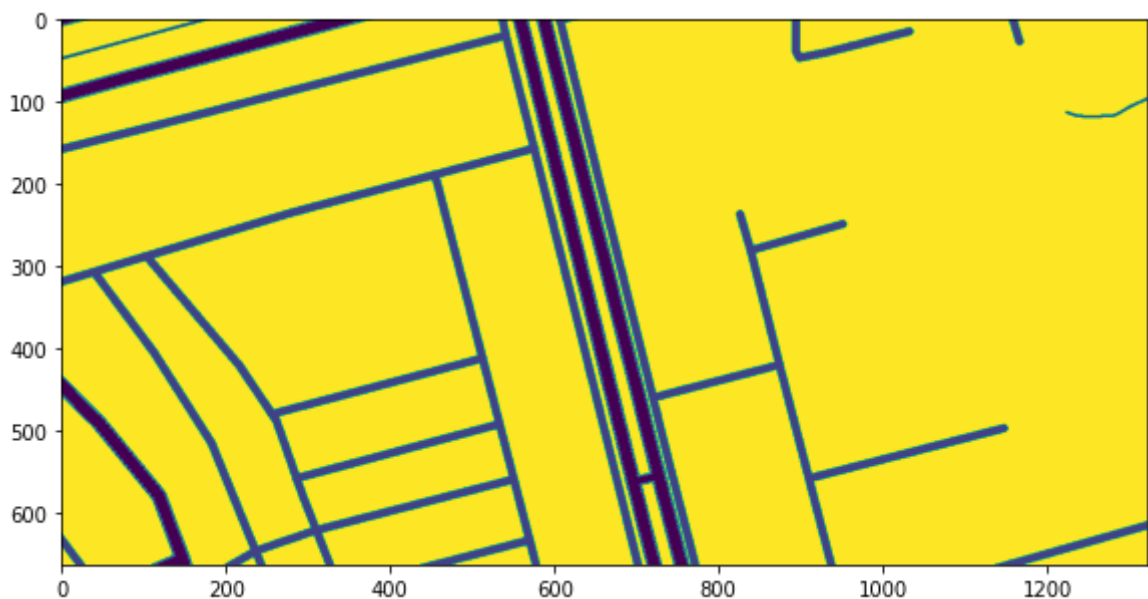
def calculate_distance(points):
    dist=math.sqrt((points[0][0]-points[1][0])**2+(points[0][1]-points[1][1]))
    return dist
```

```
In [260... segmented = road_segmentation(map_img)
cX, cY = get_point(point_img)
```

Segmented Road

```
In [261... plt.imshow(segmented)
```

```
Out[261... <matplotlib.image.AxesImage at 0x7fe5b1af7be0>
```



```
In [262... coord_points = get_two_points(cX,cY,segmented)
```

```
print(f"The estimated width of this particular section is: {calculate_distance})
```

The estimated width of this particular section is: 9.83 meter

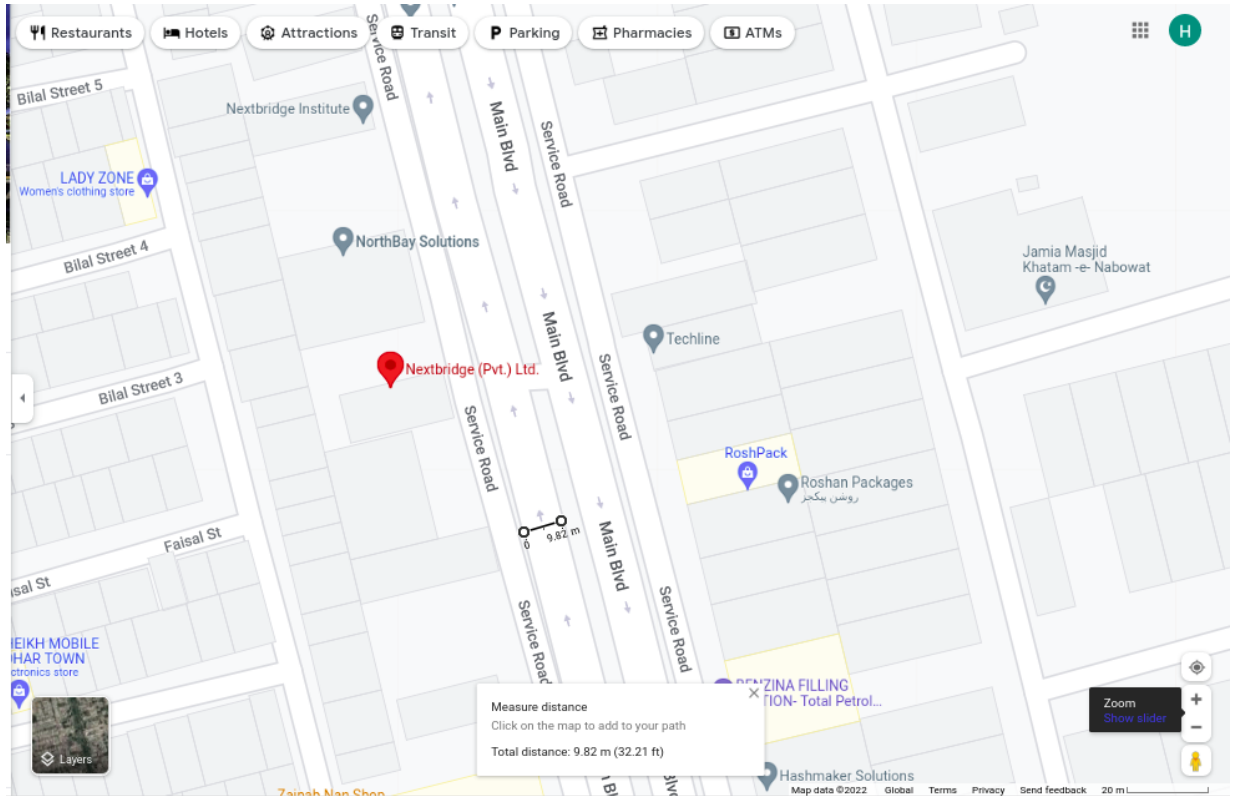
Width calculation by Google Tools

It shows the width to be 9.82

In [264...

```
from IPython.display import Image  
Image(filename='nextbridge.png')
```

Out[264...



In []: