DL Project Update 1 Code

In [1]:
```
from pathlib import Path
import re
import matplotlib.pyplot as plt
import torch
import torch.nn
import torchvision
from torch.utils.data import Dataset
import torch.nn as nn
import numpy as np
import warnings
warnings.filterwarnings("ignore")
from torchvision.transforms import ToPILImage
from IPython.display import HTML, display
```

In [2]:
```
# Creating a CustomDataset class which retrives the images and annotations.

class CustomDataset(Dataset):
    def __init__(self,path_of_image,path_of_maskedimage):
        super().__init__()
        images_path = Path(path_of_image)
        maskedimages_path = Path(path_of_maskedimage)
        self.images = [p for p in images_path.glob('*.jpg')]
        self.maskedimages = [p for p in maskedimages_path.glob('*.png')]
        self.transform1 = torchvision.transforms.Compose([torchvision.transforms.Re
                                                          torchvision.transforms.Nor
        self.transform2 = torchvision.transforms.Compose([torchvision.transforms.Re
                                                          torchvision.transforms.Nor

    def __len__(self):
        length = len(self.images)
        return length

    def __getitem__(self,index):
        img = torchvision.io.read_image(str(self.images[index]))
        masked_img = torchvision.io.read_image(str(self.maskedimages[index]))
        img = torch.tensor(img,dtype=torch.float)
        masked_img = torch.tensor(masked_img,dtype=torch.float)
        img = self.transform1(img)
        masked_img = self.transform2(masked_img)
        return img,masked_img
```

In [3]:
```
# Creating a tuple of Training, Validation and Testing Datasets.
d = (CustomDataset('Dataset_train_val_test/Images/train','Dataset_train_val_test/an
```

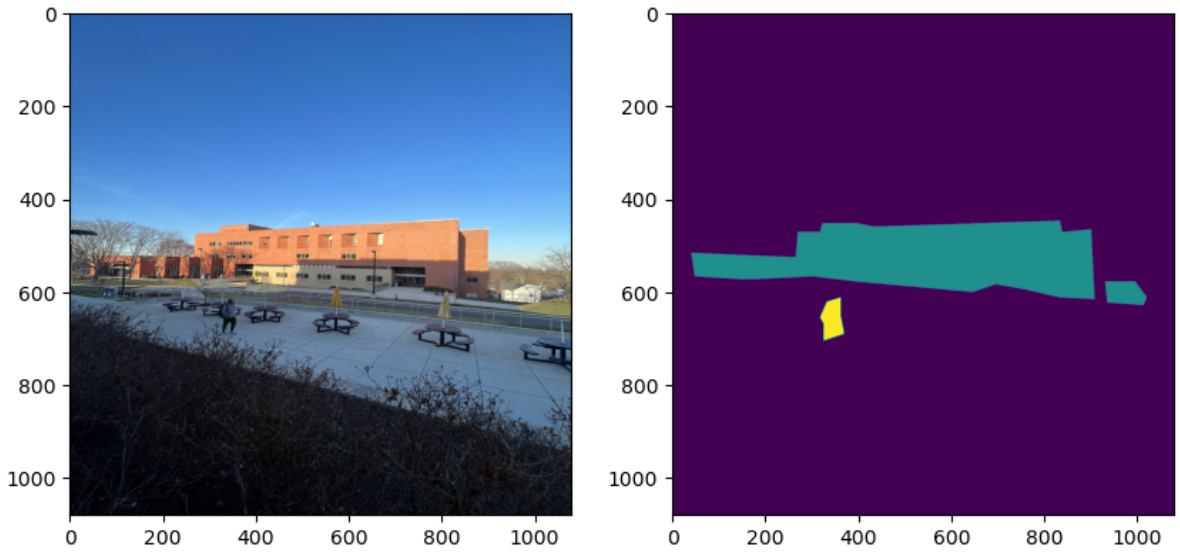
In [4]:
```
def print_with_font_size(text, font_size=5):
    display(HTML(f"<font size='{font_size}'>{text}</font>"))
```

In [5]:
```
for i in range(0,3):
    dataset = d[i]
```

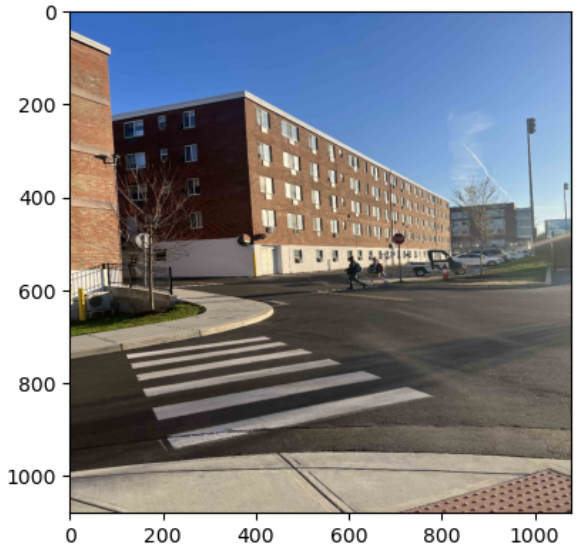
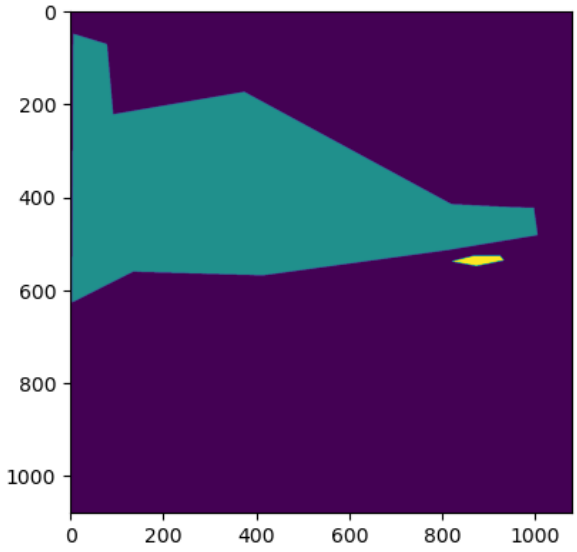
```
    train_dataloader = torch.utils.data.DataLoader(dataset,batch_size=8)
    if i == 0 :
        print_with_font_size("Images and Annotated Images of Training Images : ", f
    elif i == 1 :
        print_with_font_size("Images and Annotated Images of Validation Images : ",
    else :
        print_with_font_size("Images and Annotated Images of Testing Images : ", fo
    for batch in train_dataloader:
        imgs,masked_imgs = batch
        img_np = imgs[0].permute([1,2,0]).numpy()
        maskedimage_np = masked_imgs[0].permute([1,2,0]).numpy()
        fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(10, 5))
        ax1.imshow(img_np)
        ax2.imshow(maskedimage_np)
        plt.show()
```

Images and Annotated Images of Training Images :

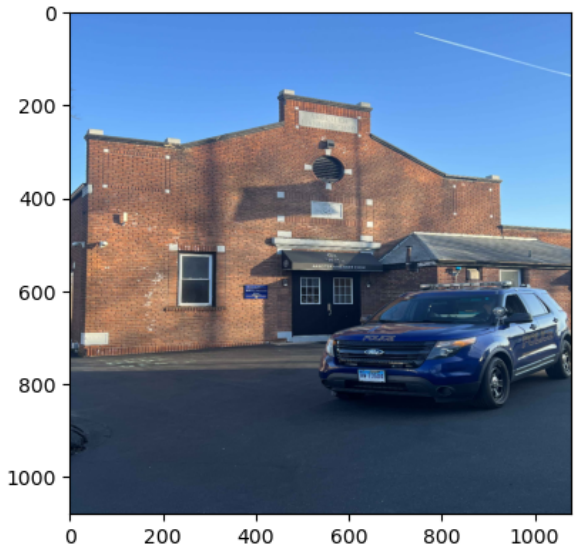
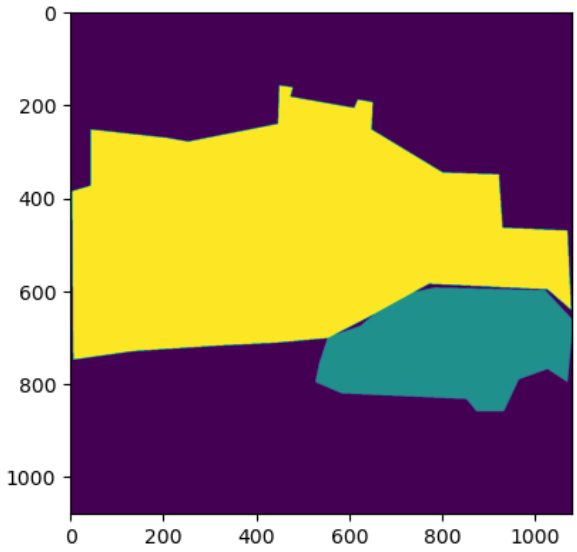
```
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).
```



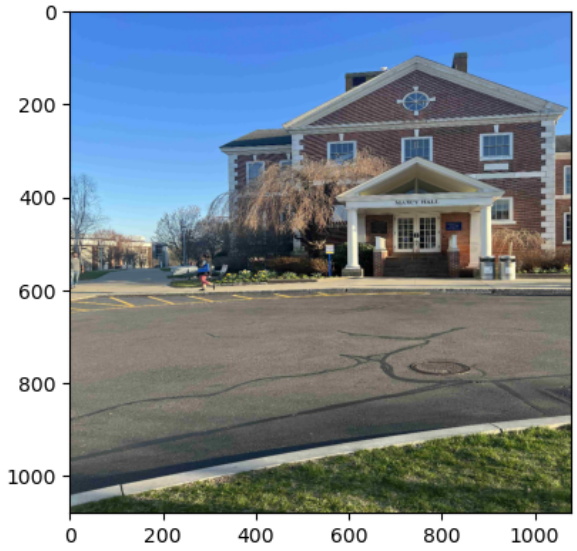
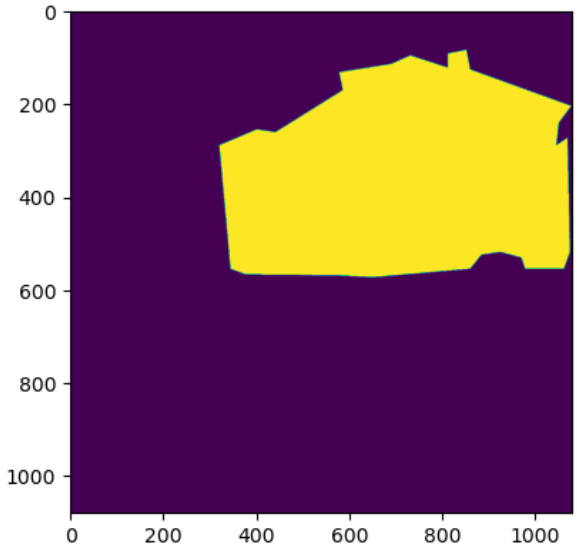

```
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).
```

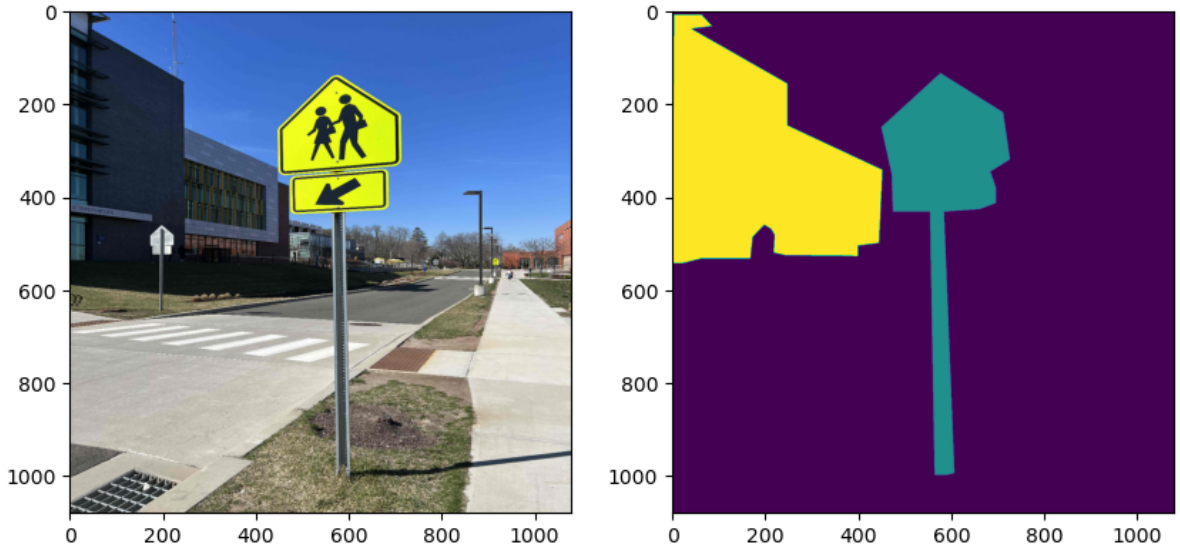
```
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).
```

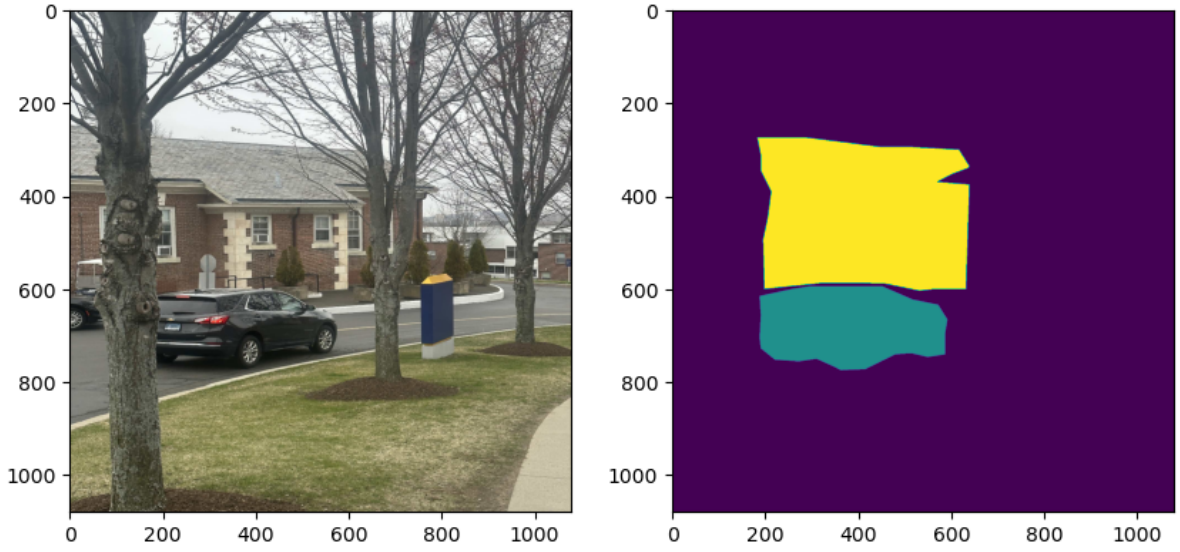
```
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).
```

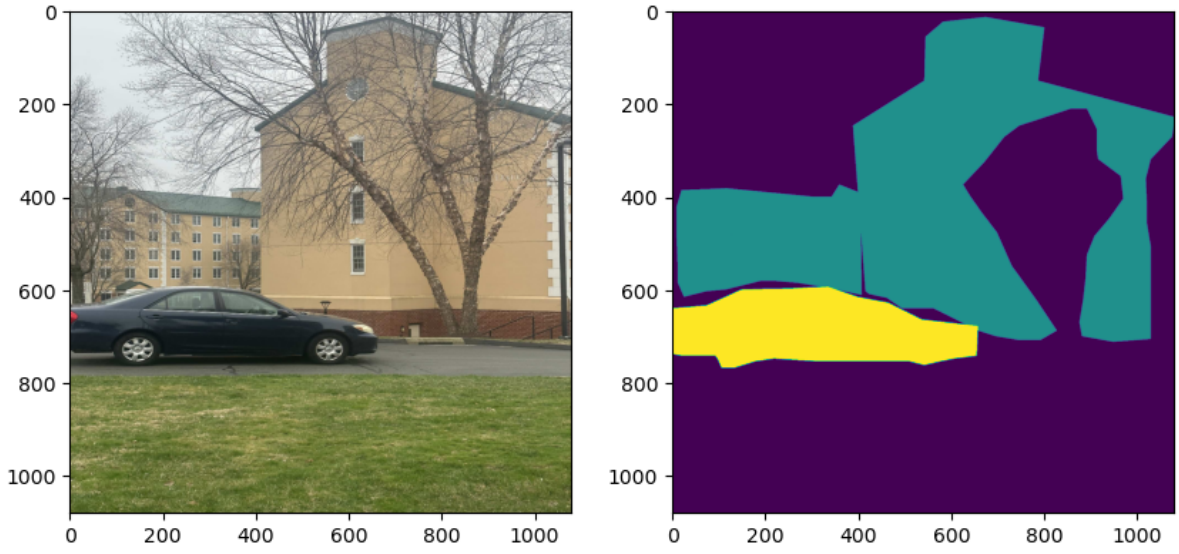
```
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).
```



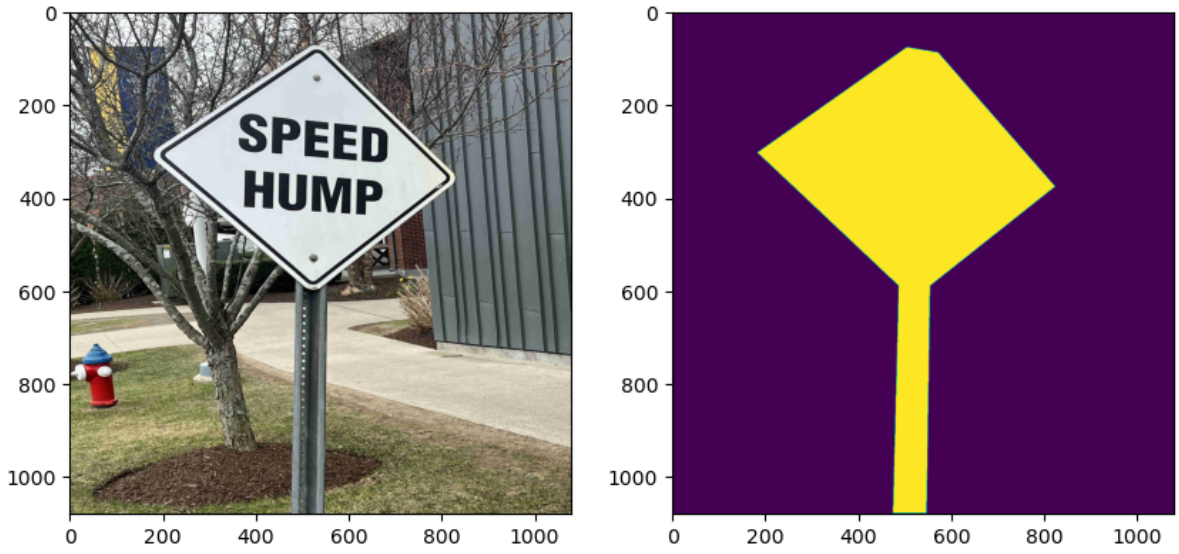

```
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).
```



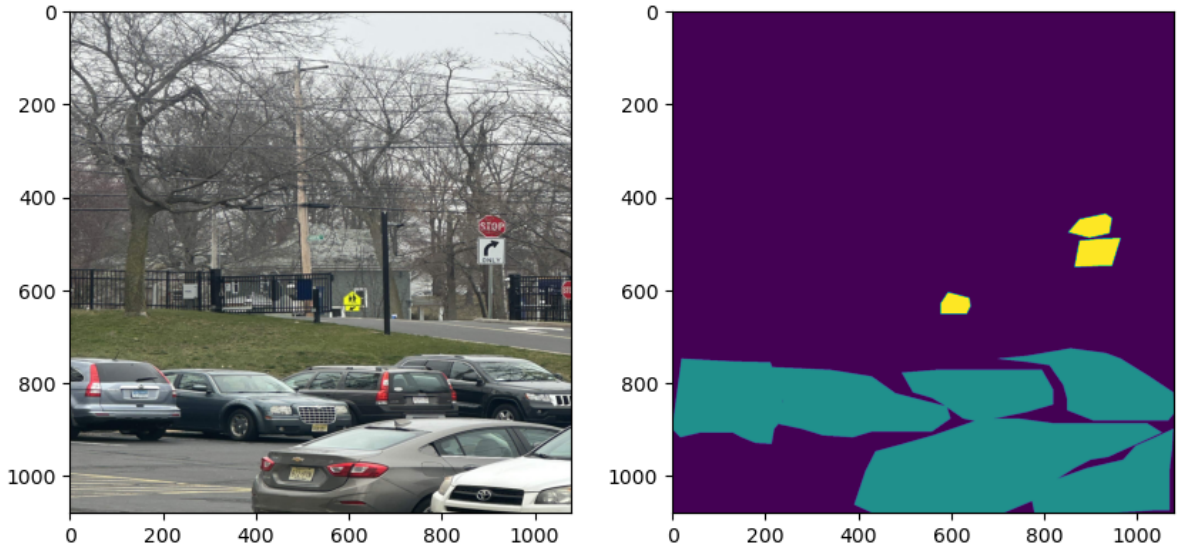

```
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).
```



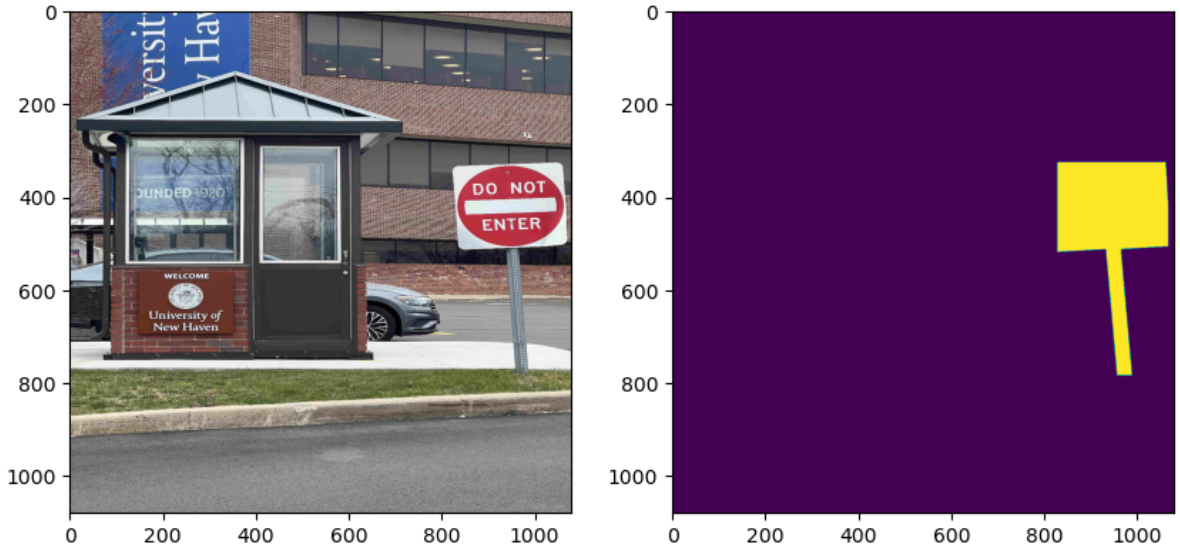

```
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).
```



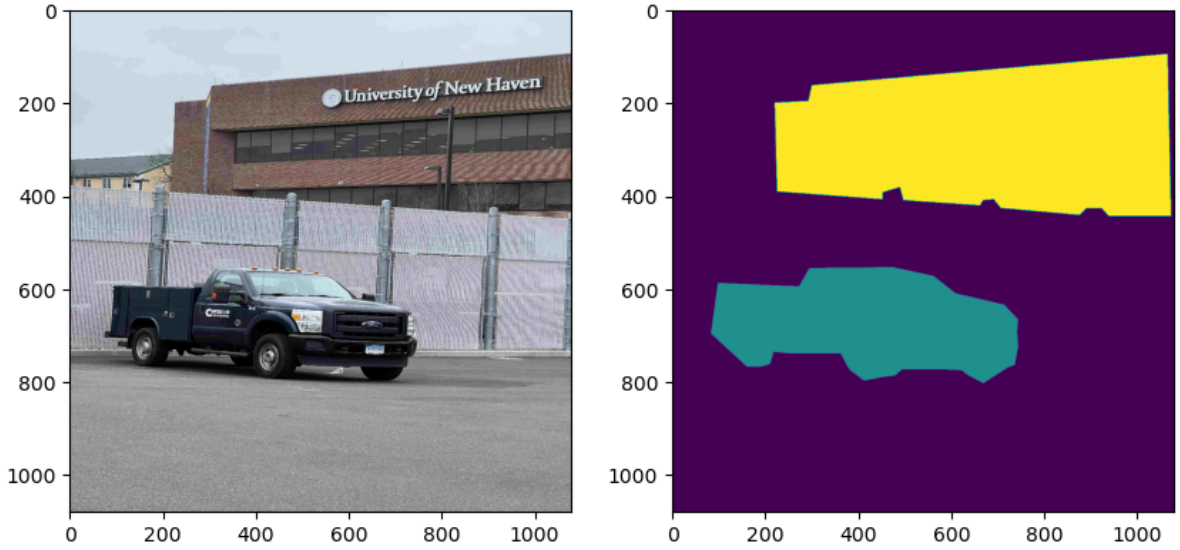

```
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).
```



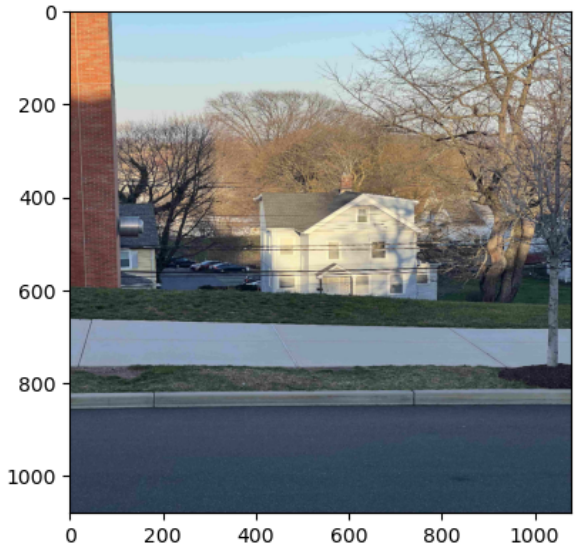
```
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).
```

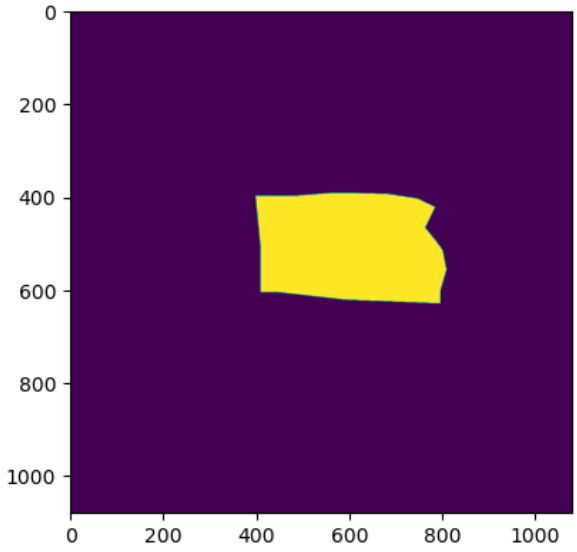



```
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).
```

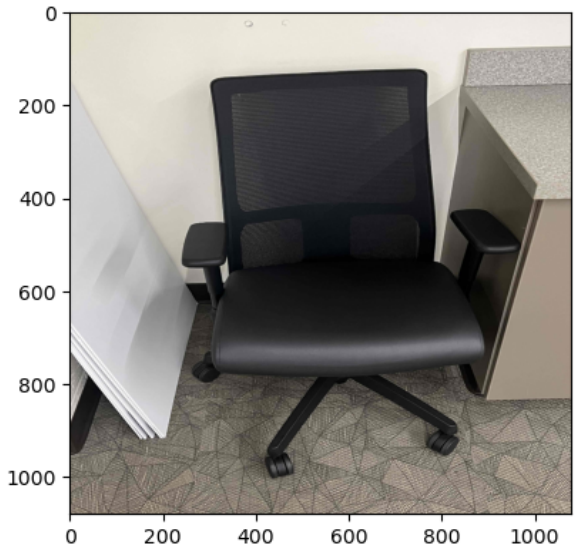


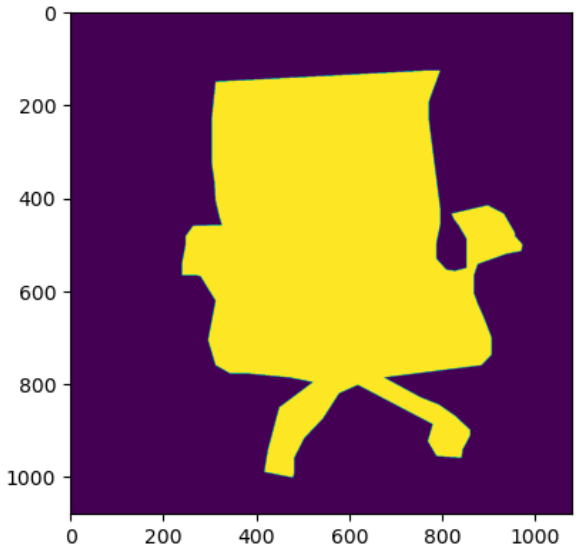

```
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).
```





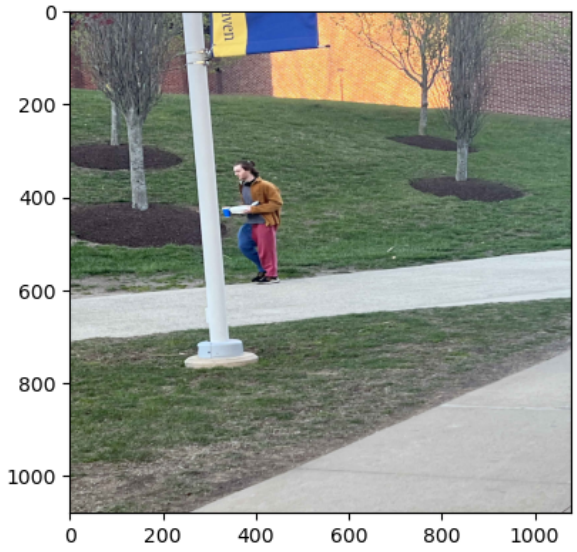

```
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).
```

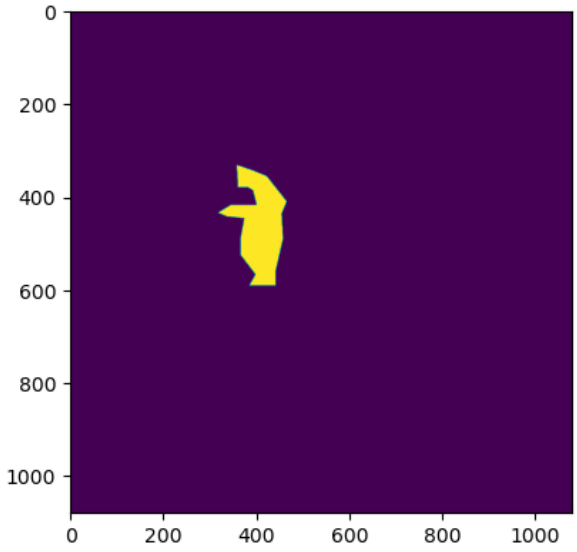




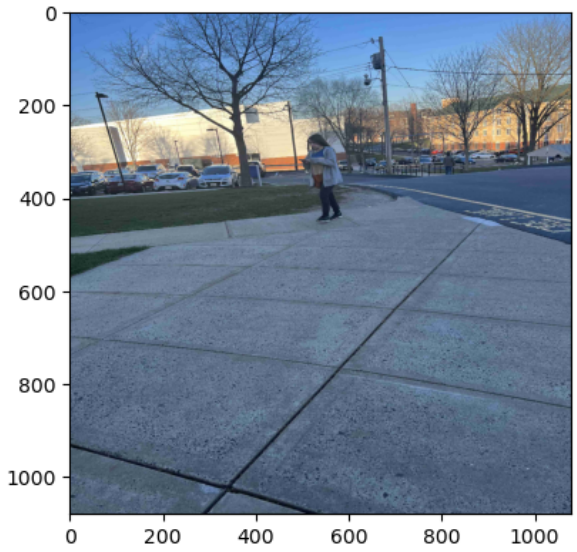
Images and Annotated Images of Validation Images :

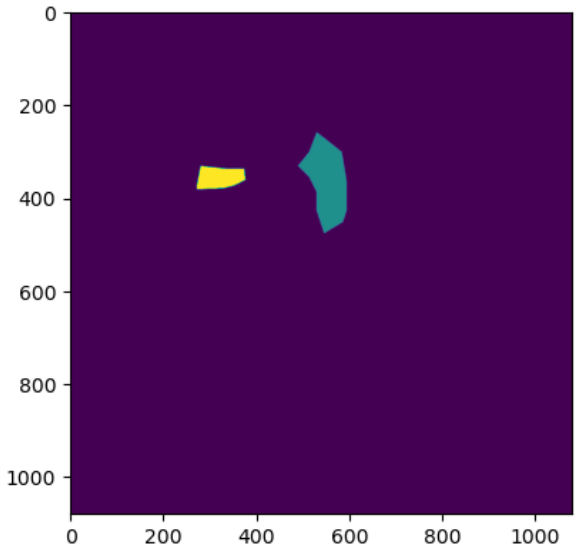
```
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).
```





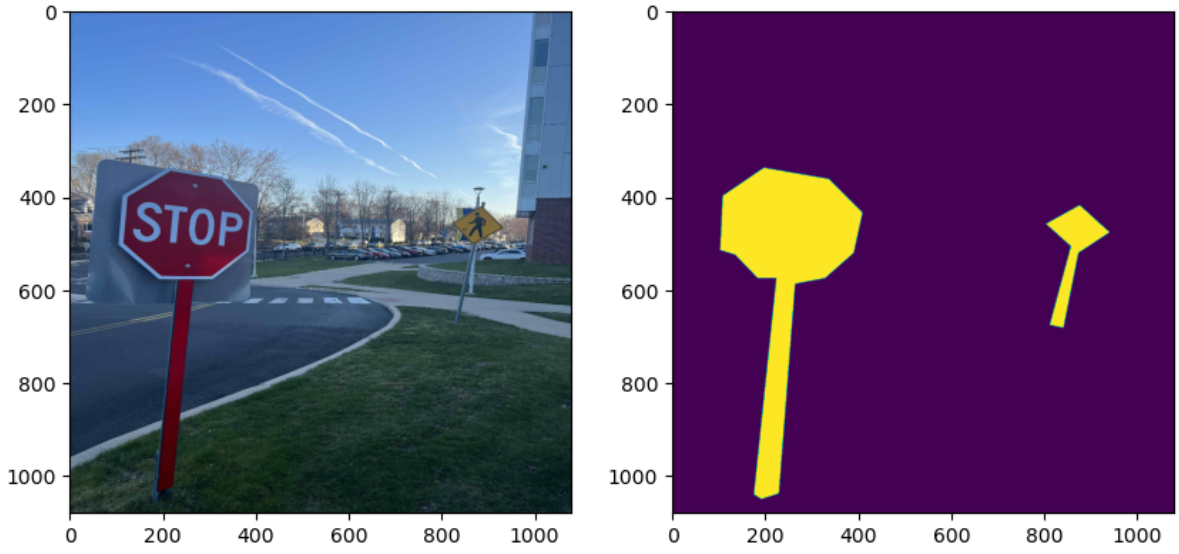

```
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).
```



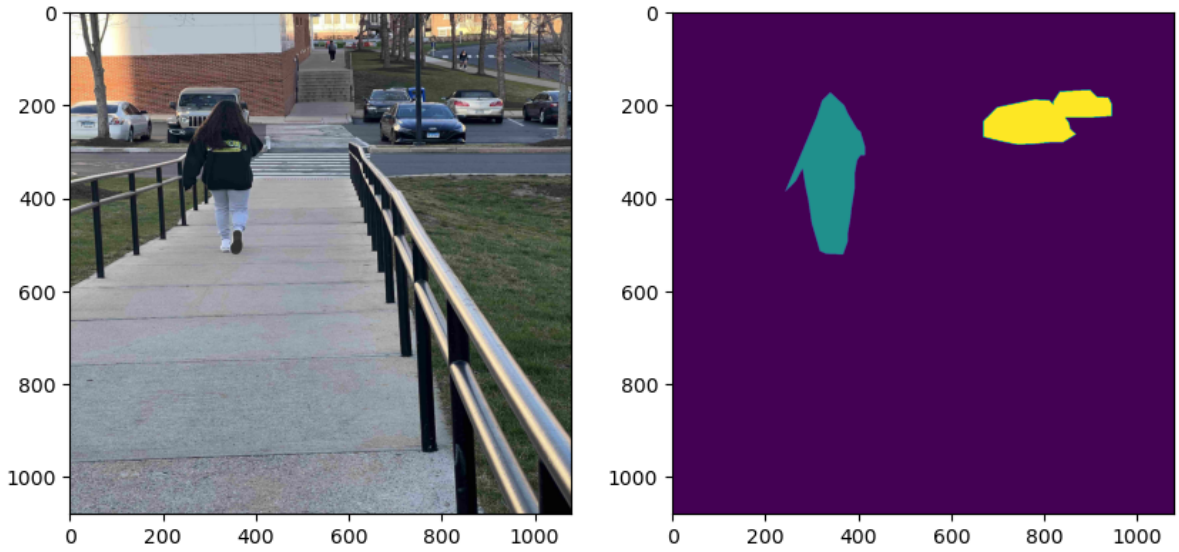


Images and Annotated Images of Testing Images :

```
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).
```




```
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).
```



In [ ]: