```
Train epoch: 9984, MSE_Loss: 1.3624165377
Train epoch: 9985, MSE_Loss: 1.3764658528
Train epoch: 9986, MSE_Loss: 1.3777410260
Train epoch: 9987, MSE_Loss: 1.3841593755
Train epoch: 9988, MSE_Loss: 1.8186403556
Train epoch: 9989, MSE_Loss: 1.3619352664
Train epoch: 9990, MSE_Loss: 1.3760128255
Train epoch: 9991, MSE_Loss: 1.8334690460
Train epoch: 9992, MSE_Loss: 1.3680346097
Train epoch: 9993, MSE_Loss: 1.3678983791
Train epoch: 9994, MSE_Loss: 1.3826696702
Train epoch: 9995, MSE_Loss: 1.8357958708
Train epoch: 9996, MSE_Loss: 1.4202316872
Train epoch: 9997, MSE_Loss: 1.4369695229
Train epoch: 9998, MSE_Loss: 1.4368388355
Train epoch: 9999, MSE_Loss: 1.4230961289
Train epoch: 10000, MSE_Loss: 1.4067226989
Finished Training
```

In [5]:

```
PATH = './SegModelEP%s.pth' %(n_epoch)
torch.save(net.state_dict(), PATH)
```

In [6]:

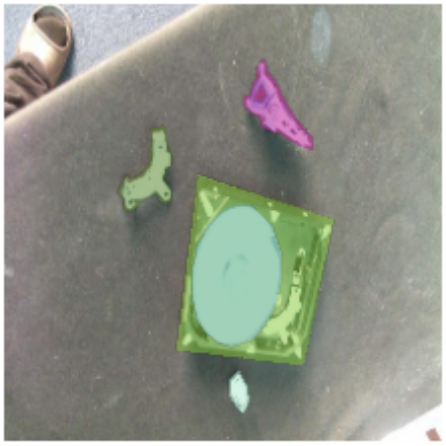
```
#train results
with torch.no_grad():
    for i, data in enumerate(trainLoader, 0):
        inputs, labels = data[0].to(device), data[1].to(device)
        inputs = inputs.float()
        labels = labels.float()
        outputs = net(inputs)
        outputs = outputs.cpu().data.numpy()
        inputs, labels = data[0].cpu().data.numpy(), data[1].cpu().data.numpy()

        for j in range(len(outputs)):
            suboutputs = outputs[j]
            suboutputs[suboutputs>0]=1
            suboutputs = suboutputs.astype('uint8')
            plotSegmentation(inputs[j]*255, suboutputs, 'test')
            #break
        break
```

```
Resnet50 prediction:
hard_disk with prediction0.9965 probabilities
disk with prediction0.9987 probabilities
chip with prediction0.987 probabilities
reader with prediction0.9798 probabilities
```

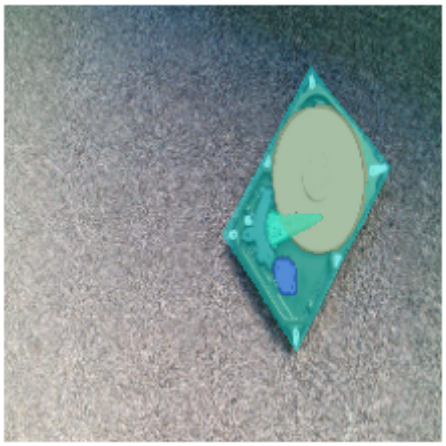
```
y_part with prediction0.96 probabilities
GoogLeNet prediction:
hard_disk with prediction0.8514 probabilities
disk with prediction0.9375 probabilities
chip with prediction0.7505 probabilities
reader with prediction0.8085 probabilities
y_part with prediction0.4647 probabilities
```

```
Resnet50 prediction:
hard_disk with prediction0.9978 probabilities
disk with prediction0.9988 probabilities
chip with prediction0.9722 probabilities
reader with prediction0.9536 probabilities
y_part with prediction0.9834 probabilities
GoogLeNet prediction:
hard_disk with prediction0.862 probabilities
disk with prediction0.9067 probabilities
chip with prediction0.7955 probabilities
reader with prediction0.8259 probabilities
y_part with prediction0.4721 probabilities
```



```
Resnet50 prediction:
hard_disk with prediction0.9976 probabilities
disk with prediction0.9982 probabilities
chip with prediction0.9743 probabilities
reader with prediction0.9907 probabilities
y_part with prediction0.9727 probabilities
GoogLeNet prediction:
hard_disk with prediction0.7103 probabilities
disk with prediction0.9506 probabilities
chip with prediction0.5064 probabilities
reader with prediction0.6633 probabilities
y_part with prediction0.7084 probabilities
```

```
Resnet50 prediction:
hard_disk with prediction0.9968 probabilities
disk with prediction0.9722 probabilities
chip with prediction0.3196 probabilities
reader with prediction0.961 probabilities
y_part with prediction0.8802 probabilities
GoogLeNet prediction:
hard_disk with prediction0.692 probabilities
disk with prediction0.7425 probabilities
chip with prediction0.2669 probabilities
reader with prediction0.3017 probabilities
y_part with prediction0.3434 probabilities
```



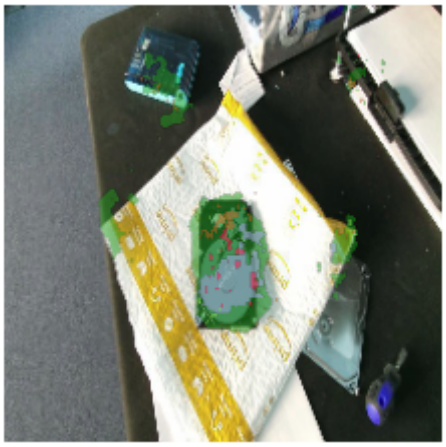
```
Resnet50 prediction:
hard_disk with prediction0.9981 probabilities
disk with prediction0.9991 probabilities
chip with prediction0.9904 probabilities
reader with prediction0.9774 probabilities
y_part with prediction0.981 probabilities
GoogLeNet prediction:
hard_disk with prediction0.825 probabilities
disk with prediction0.8835 probabilities
chip with prediction0.8409 probabilities
reader with prediction0.7525 probabilities
y_part with prediction0.6624 probabilities
```

In [15]:

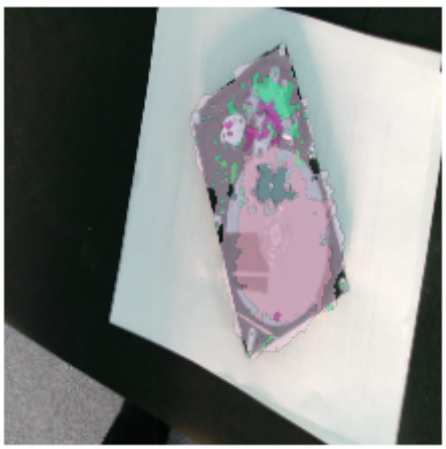
```
#Test results
with torch.no_grad():
    for i, data in enumerate(testLoader, 0):
        inputs, labels = data[0].to(device), data[1].to(device)
        inputs = inputs.float()
        labels = labels.float()
        outputs = net(inputs)
```

```
        outputs = outputs.cpu().data.numpy()
        inputs, labels = data[0].cpu().data.numpy(), data[1].cpu().data.numpy()

        for j in range(len(outputs)):
            suboutputs = outputs[j]
            suboutputs[suboutputs>0]=1
            suboutputs = suboutputs.astype('uint8')
            plotSegmentation(inputs[j]*255, suboutputs, 'test')
            #break
        #break
```

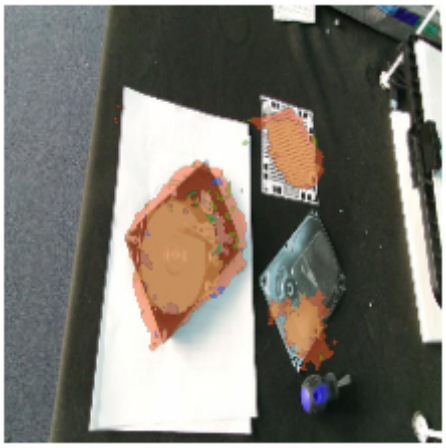


```
Resnet50 prediction:
hard_disk with prediction0.1056 probabilities
disk with prediction0.5468 probabilities
chip with prediction0.4246 probabilities
reader with prediction0.0516 probabilities
y_part with prediction0.037 probabilities
GoogLeNet prediction:
hard_disk with prediction0.2264 probabilities
disk with prediction0.1536 probabilities
chip with prediction0.255 probabilities
reader with prediction0.1918 probabilities
y_part with prediction0.0548 probabilities
```



```
Resnet50 prediction:
hard_disk with prediction0.9602 probabilities
disk with prediction0.5298 probabilities
chip with prediction0.1671 probabilities
reader with prediction0.0055 probabilities
y_part with prediction0.4857 probabilities
GoogLeNet prediction:
hard_disk with prediction0.3162 probabilities
disk with prediction0.1696 probabilities
```

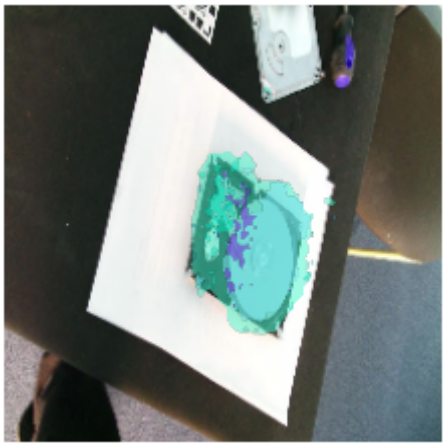
```
chip with prediction0.3669 probabilities
reader with prediction0.1905 probabilities
y_part with prediction0.2293 probabilities
```



```
Resnet50 prediction:
hard_disk with prediction0.8806 probabilities
disk with prediction0.9679 probabilities
chip with prediction0.6058 probabilities
reader with prediction0.3597 probabilities
y_part with prediction0.0365 probabilities
GoogLeNet prediction:
hard_disk with prediction0.2386 probabilities
disk with prediction0.3417 probabilities
chip with prediction0.2836 probabilities
reader with prediction0.2626 probabilities
y_part with prediction0.1526 probabilities
```

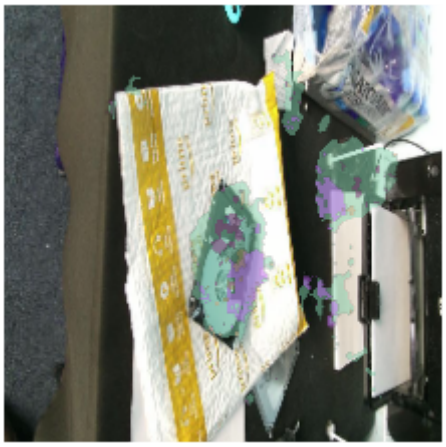
```
Resnet50 prediction:
hard_disk with prediction0.9746 probabilities
disk with prediction0.994 probabilities
chip with prediction0.9579 probabilities
reader with prediction0.9374 probabilities
y_part with prediction0.6829 probabilities
GoogLeNet prediction:
hard_disk with prediction0.7066 probabilities
disk with prediction0.5923 probabilities
chip with prediction0.6931 probabilities
reader with prediction0.594 probabilities
y_part with prediction0.4843 probabilities
```

```
Resnet50 prediction:
hard_disk with prediction0.8205 probabilities
disk with prediction0.9725 probabilities
chip with prediction0.5325 probabilities
reader with prediction0.7978 probabilities
y_part with prediction0.7346 probabilities
GoogLeNet prediction:
hard_disk with prediction0.5061 probabilities
disk with prediction0.6796 probabilities
chip with prediction0.4583 probabilities
reader with prediction0.2861 probabilities
y_part with prediction0.1589 probabilities
```



```
Resnet50 prediction:
hard_disk with prediction0.0251 probabilities
disk with prediction0.8894 probabilities
chip with prediction0.5362 probabilities
reader with prediction0.0562 probabilities
y_part with prediction0.956 probabilities
GoogLeNet prediction:
hard_disk with prediction0.3063 probabilities
disk with prediction0.3733 probabilities
chip with prediction0.2278 probabilities
reader with prediction0.3436 probabilities
y_part with prediction0.0701 probabilities
```

```
Resnet50 prediction:
hard_disk with prediction0.9877 probabilities
disk with prediction0.9906 probabilities
chip with prediction0.951 probabilities
reader with prediction0.8567 probabilities
y_part with prediction0.9717 probabilities
GoogLeNet prediction:
hard_disk with prediction0.6658 probabilities
disk with prediction0.7026 probabilities
chip with prediction0.5363 probabilities
reader with prediction0.4164 probabilities
y_part with prediction0.2497 probabilities
```



```
Resnet50 prediction:
hard_disk with prediction0.0816 probabilities
disk with prediction0.6118 probabilities
chip with prediction0.5697 probabilities
reader with prediction0.0481 probabilities
y_part with prediction0.4575 probabilities
GoogLeNet prediction:
hard_disk with prediction0.2372 probabilities
disk with prediction0.1196 probabilities
chip with prediction0.355 probabilities
reader with prediction0.2208 probabilities
y_part with prediction0.2532 probabilities
```

In [ ]: