A. Supplementary Material

A.1. Transferable Examples for Semantic Segmentation and Object Detection

As shown in the main paper, adversarial examples can be transformed across networks with different training data. based on different architectures, and even for different tasks. Some typical examples are shown in Figure 1, where the adversarial examples from Network 1 are able to transfer to Network 2.

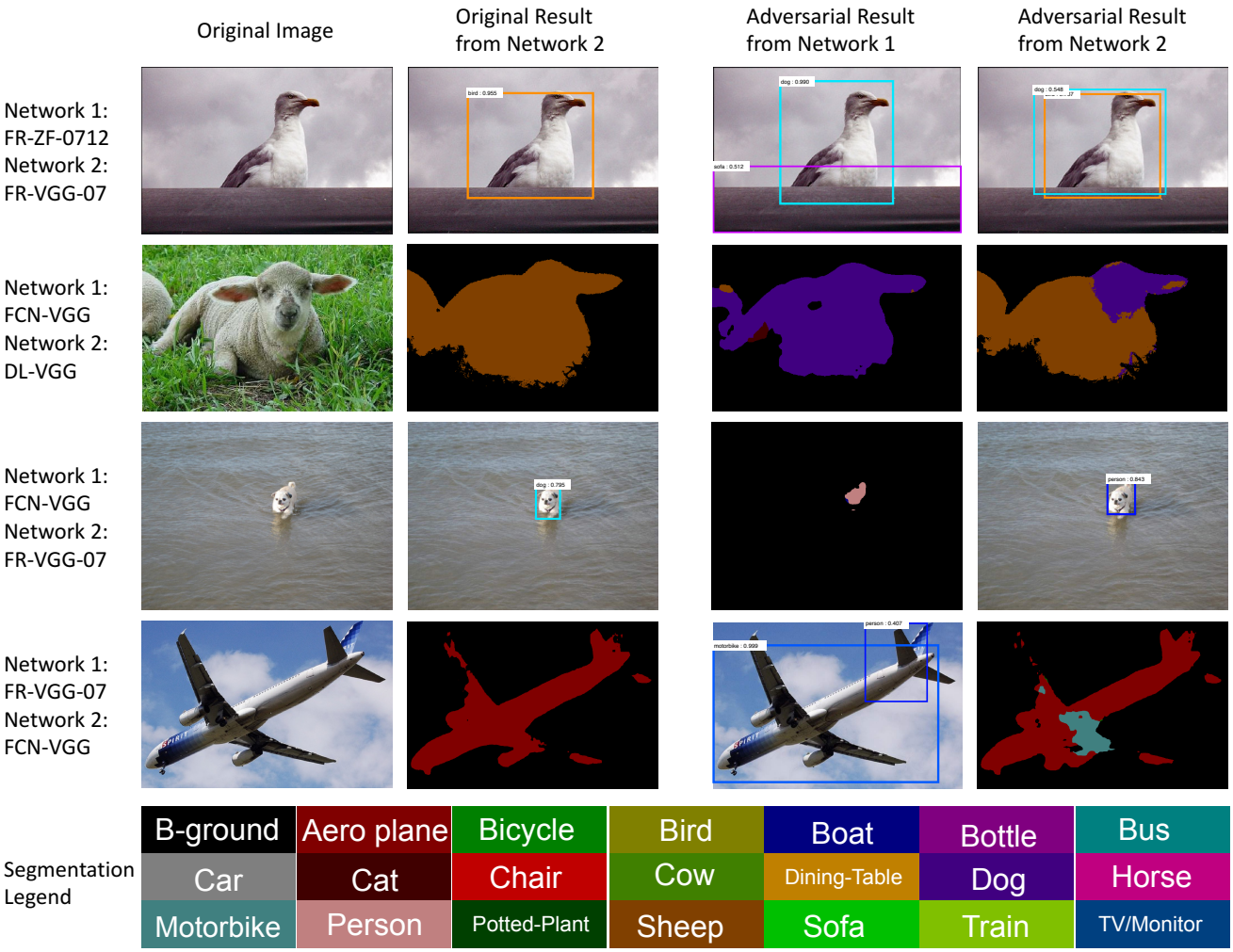

Figure 1. Transferrable examples for semantic segmentation and object detection. These four rows, from top to bottom, shows the adversarial attack examples within two detection networks, within two segmentation networks, using a segmentation network to attack a detection network and in the opposite direction. The segmentation legend borrows that in [1].

A.2. Generating Geometric Patterns

As an additional showcase, the deep segmentation networks can be confused to output some geometric shapes, including *stripes*, *circles*, *triangles*, *squares*, *etc.*, after different adversarial perturbations is added to the original image. Results are shown in Figure 2. Here, the added adversarial perturbation varies from case to case.

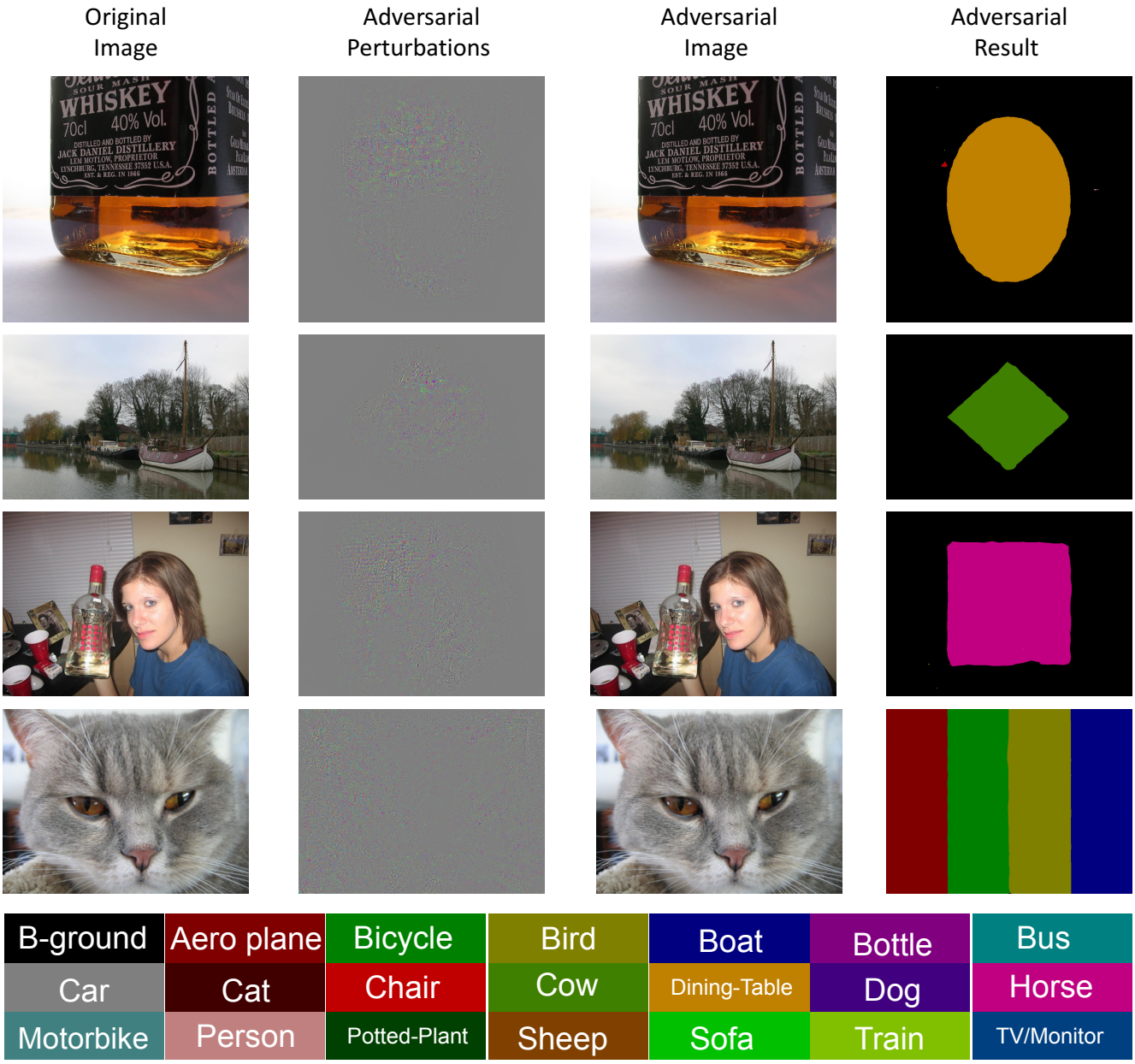

Figure 2. The adversarial perturbations confuse the deep networks to output different geometric patterns as segmentation results, such as a circle (the first row), a diamond (the second row), a square (the third row), and stripes (the fourth row). Here, **FCN-Alex** is used as the baseline network (defender). All the perturbations are **magnified by 10** for better visualization. The segmentation legend borrows that in [1].

A.3. Same Noise, Different Outputs

In Figure 2 of the main article, we show that we can generate some adversarial perturbations to make a deep segmentation network output a pre-specified segmentation mask (*e.g.*, `ICCV` and `2017`). But, the perturbations used to generate these two segmentation masks are different.

Here, we present a more challenging task, which uses the same perturbations to confuse two networks. More specifically, we hope to generate a perturbation $\mathbf{r}$, when it is added to an image $\mathbf{X}$, the **FCN-Alex** and the **FCN-VGG** models are confused to output `ICCV` and `2017`, respectively. To implement this, we apply the locally linear property of the network, and add two sources of perturbations, *i.e.*, $\mathbf{r} = \mathbf{r}_1 + \mathbf{r}_2$, where $\mathbf{r}_1$ is generated on **FCN-Alex** with the mask `ICCV`, and $\mathbf{r}_2$ is generated on **FCN-VGG** with the mask `2017`. As shown in Figure 3, our simple strategy works very well, although the segmentation boundary of each letter or digit becomes somewhat jagged.

Figure 3. We add one adversarial perturbation (**magnified by 10**) to the same original image to generate different pre-specified segmentation masks on two deep segmentation networks (**FCN-Alex** and **FCN-VGG**). This is a more difficult task compared to that shown in Figure 2 of the main article, where two different adversarial perturbations are used to generate two pre-specified segmentation masks. The blue regions in the segmentation masks are predicted as *bus*, a randomly selected class.

References

[1] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. Torr. Conditional random fields as recurrent neural networks. In *International Conference on Computer Vision*. IEEE, 2015.