Technical Documents

1 Team information

Team members: Chao Du, Tianyu Pang, Yinpeng Dong

Team track: Defense against adversarial attack

Team names: wukong sun (rank 2); heshang sha (rank 4); seng tang (rank 5); bajie zhu (rank 6)

2 Methods

2.1 Basic model

In the submission of team name: *bajie zhu*, we apply our basic model, as described in Figure 1. The model can be implemented in two steps:

Step 1: The input image is fed into four different denoisers, and then get four denoised output images.

Step 2: The four denoised output images are separately fed into four following classifiers, and the four predicted logits are averaged to make final predictions.

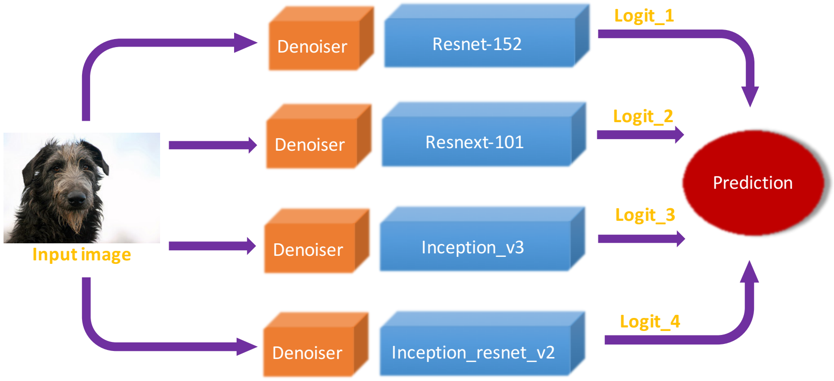

Figure 1: Flow diagram of the basic model.

2.2 Advanced models

In advanced models, we add extra enhanced blocks based on the basic model.

2.2.1 Uniform random noise

In the submission of team name: *wukong sun*, we add uniform random noise on the denoised output images, and then feed them into classifiers. Note that the pixel values of image are normalized into the interval of $[-1, 1]$. In our experiments, we find that when adding noise that uniformly distributes on $[-0.25, 0.25]$, the validation error rate is the lowest. The flow diagram is shown in Figure 2.

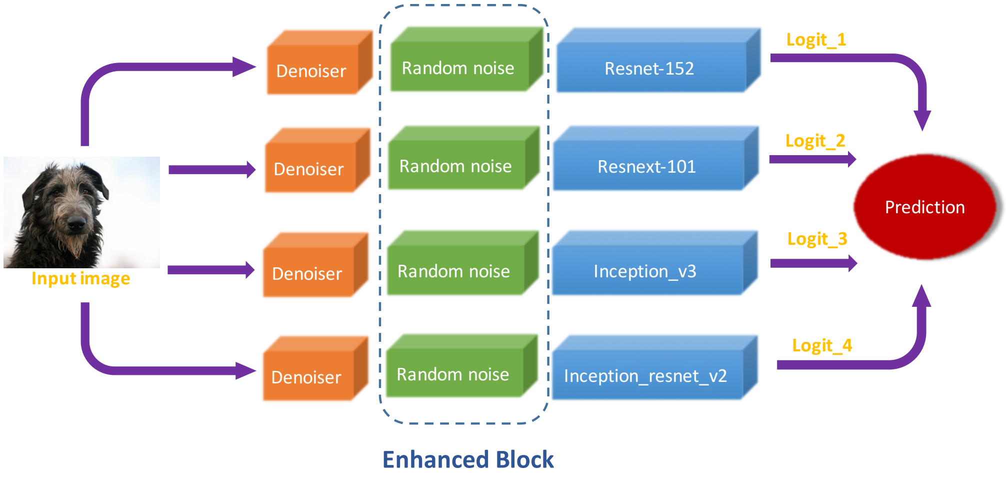

Figure 2: Flow diagram of the advanced model with uniform random noise.

2.2.2 Recurrent denoiser

In the submission of team name: *heshang sha*, we apply recurrent denoiser. In general, a denoiser is only passed once in order to eliminate adversarial noise, and this is analogical to the case when attacking with one-step method, e.g., FGSM. However, iterative method like I-FGSM or PGD usually perform better than one-step method, and this phenomenon inspires us to use recurrent denoiser to iteratively eliminate adversarial noise. Here recurrent denosier has a similar architecture as a recurrent nerual network (RNN), which convert the denoising process into a iterative frame.The flow diagram is shown in Figure 3.

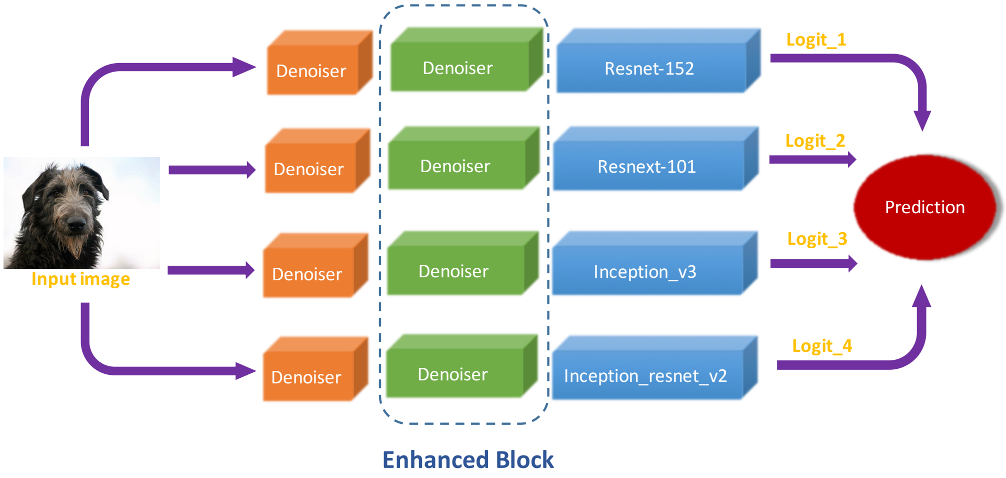

Figure 3: Flow diagram of the advanced model with recurrent denoiser.

2.2.3 Pixel shrinkage

In the submission of team name: *seng tang*, we apply a new data augmentation method, called pixel shrinkage. Pixel shrinkage is easy to implement by simply scale the pixel values with a shrinkage factor. This can be intuitively illustrated in Figure 4. Assume there is an adversarial example, which is explicit since we can directly observe it. Then the adversarial example corresponds to an implicit normal example, and induces a implicit adversarial perturbation. Here we first select a reference point, and then mix up the pixel values between adversarial example and reference point. For example, in our method we choose the origin as the reference point, and the pixel mixup becomes pixel shrinkage. According to basic theorems of similar triangles, the implicit adversarial perturbation will also shrink with the same shrinkage factor. The flow diagram is shown in Figure 5

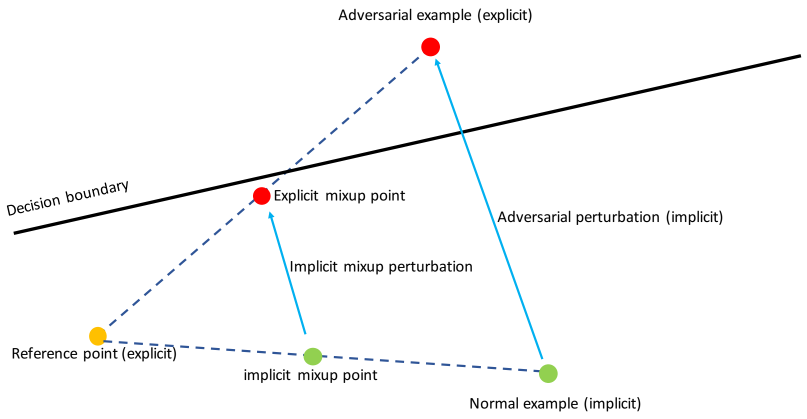
Adversarial example (explicit)
Decision boundary
Explicit mixup point
Adversarial perturbation (implicit)
Implicit mixup perturbation
Reference point (explicit)
implicit mixup point
Normal example (implicit)

Figure 4: Flow diagram of the advanced model with pixel shrinkage.

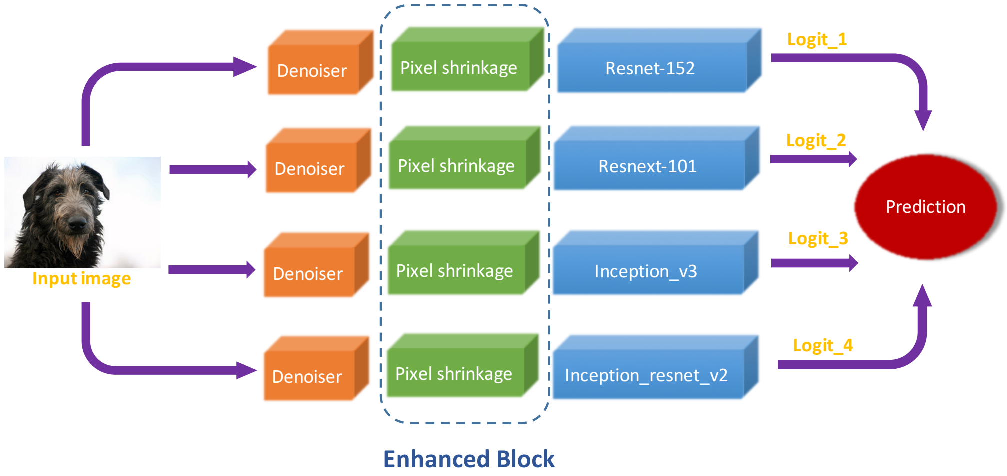

Figure 5: Flow diagram of the advanced model with pixel shrinkage.