A EXPERIMENTAL RESULT WITHOUT THE INCORPORATION OF GAUSSIAN NOISE

To determine whether the appearance of human-identifiable features is inherent to adversarial perturbations, rather than an artifact of Gaussian noise introduced in the MM+G setting, we carried out a noise-free experiment. In this test, we averaged perturbations from multiple models (referred to as the MM setting) under untargeted attack algorithms. We then repeated several experiments initially conducted in the MM+G setting to see if comparable results were obtained. Due to a limited number of source models, this investigation was limited to the ImageNet dataset.

A.1 EMERGENCE OF HUMAN-IDENTIFIABLE FEATURES

As illustrated in Figure 7, our visual analysis reveals that generated perturbations still retain human-identifiable features similar to those in the original images, which is the masking effect. Importantly, this phenomenon can be observed across the experiment, not just in the examples shown in Figure 7. However, the human-identifiable features are less pronounced in comparison to those in the MM+G setting. The number of averaged perturbations in this setting is much smaller - about a tenth compared to MM+G - thus limiting the noise reduction in the perturbations. It therefore is expected that the human-identifiable features are less obvious compared to those from MM+G.

Notably, in the case of search-based algorithms, which inherently possess randomness, additive Gaussian noise when generating perturbations is not essential, as detailed in Appendix G. Nevertheless, we still observe a masking effect in the perturbations. Our observations indicate that this masking effect persists in both gradient-based and search-based attack algorithms, even without the addition of Gaussian noise. Therefore, we can infer that the presence of human-identifiable features is inherent in the perturbations themselves, rather than being a result of added Gaussian noise.

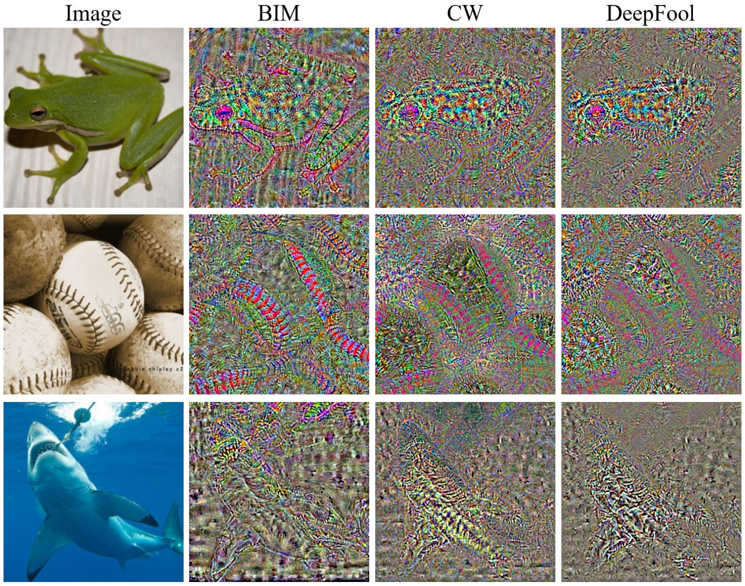

Figure 7: Adversarial perturbations generated via gradient-based algorithms in the MM setting for untargeted attacks. These perturbations maintain features that are similar to those in the original images, indicating an inherent masking effect not caused by Gaussian noise. While these features are less distinct than those in the MM+G setting, it is important to note that the number of perturbations used for averaging, in this case, is only one-tenth of that employed in the MM+G setting.

A.2 CONTOUR EXTRACTION EXPERIMENT UNDER MM SETTING

In this experiment, we replicate the contour extraction experiment described in Section 5.2.3. However, this time the perturbations are generated using the MM setting.

The experiment results are as follows:

(1) The perturbations generated from contours are a key factor causing the model to misclassify. On average, across four testing models and 200 images, under the condition of $\epsilon = 0.02$, these perturbations lower the model's accuracy rate from 81.8% to between 32.9-39.2%. In contrast, when images include the background but exclude the contour component of perturbations, the accuracy stays within the range of 65.0% to 69.8%.

(2) Figure 8 plots the average model accuracy vs. ϵ. Here, we vary the L_{inf} norm of the perturbation contours and backgrounds by adjusting ϵ from 0.01 to 0.1 in increments of 0.01, as per Eqn. 2. Note that the experimental setup mirrors that in Section 5.2.3. We can see that as ϵ increases to 0.1, the attack strength of the contours drops down to 11.3%, whereas the strength of the background attack quickly saturates at 61.4%.

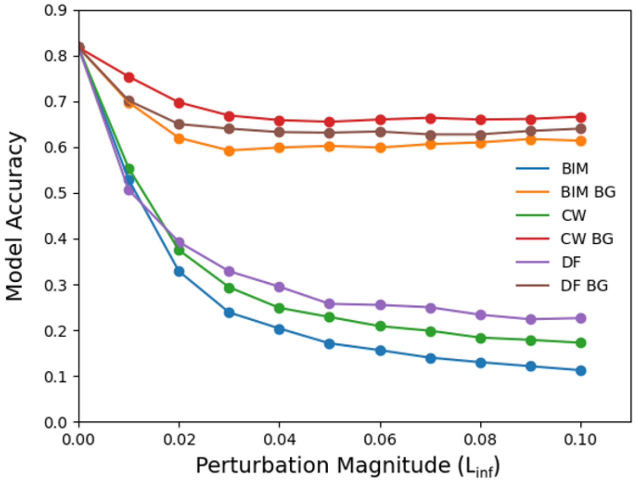

Figure 8: Effect of L_{inf} norm variation on contour and background attack strength in the MM setting. The x-axis represents ϵ, and the y-axis shows the average accuracy over four models and 200 test images. Data points labeled only by the attack algorithm denote perturbations with contours being extracted, whereas data points labeled by both the attack algorithm and 'BG' denote background extraction. As the L_{inf} norm increases, the influence of background perturbations stabilizes (61.4-66.9%), whereas contour perturbations continue to substantially degrade model accuracy, dropping to 11.3% at $\epsilon = 0.1$.

The results from the MM setting align with those from the MM+G setting, i.e., showing notably stronger attack strength for contour perturbations compared to that of the background perturbations. These findings reaffirm that the observed strong attack strength from human-identifiable features is not due to the inclusion of noise in the MM+G setting.

A.3 Convergence of Cosine Similarities

Finally, we repeat the experiment of calculating cosine similarities under the MM setting. We have found that the average cosine similarities of perturbations generated by different attack algorithms, after contour extraction, range between 0.40 and 0.58. This is higher than the cosine similarities of perturbations generated in the SM setting, ranging between 0.13 and 0.31, as shown in Figure 9. This is consistent with the results obtained from perturbations in the MM+G setting, shown in Figure 5.

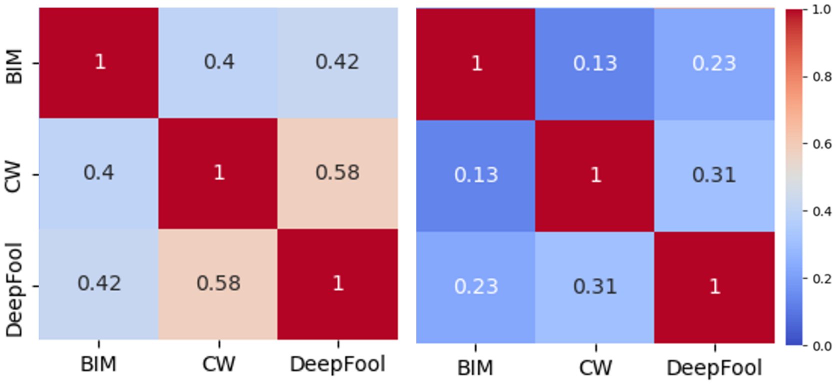

Figure 9: Measuring cosine similarity for perturbations in the MM (left panel) and SM (right panel) settings. For the MM setting, the average cosine similarities among various attack algorithms ranged from 0.40 to 0.58 after contours were extracted. Compared to the SM setting (0.13 to 0.31), these values are higher and agree with the results obtained from the MM+G setting.

Consistent findings between the MM+G and MM settings reinforce the idea that human-identifiable features are an inherent property of adversarial perturbations, rather than the consequence of added noise.

B Detailed Information for Experimental Setup

B.1 Parameters for Gradient-based Attacks

We analyze both BIM and CW attacks in both targeted and untargeted modes. The DeepFool attack is also investigated, but only in untargeted attack mode, as it does not support targeted attacks. BIM attack uses ϵ = 0.02, α = 0.0008, and 50 iterations. The CW attack is configured with c=5, κ=5, and 1,000 iterations. For the DeepFool attack, we set η to be 0.02 and limited the number of iterations to 50. Please note that the above notations are consistent with those used in the original papers.

Due to the computational demands of applying DeepFool to ImageNet under the MM+G setting, we modify the attack by focusing only on the top 10 classes ranked by output logits, rather than evaluating all ImageNet classes, to identify the most appropriate target label. This adjustment resulted in a hundredfold reduction in computational time. All attack implementations were carried out using Torchattacks (Kim, 2020).

B.2 Displaying Perturbations

After generating adversarial perturbations, we need to present them as images, which requires scaling. It is important to note that the mechanisms for targeted and untargeted attacks to fool neural networks are different, so the methods for displaying them also vary.

For untargeted attacks, our method is to invert the perturbation (multiplying by -1) to cancel out the negative sign brought on by the masking effect, as discussed in Section 5.1. We linearly adjust the perturbation's mean and standard deviation to match the average mean and variance of the ImageNet dataset. This step corrects for pixel value biases that cause color distortion after averaging the perturbations. It helps our observation of the masking effect and we expect the displayed perturbations to reveal human-identifiable features that resemble those from the original image.

Perturbations from targeted attacks often exhibit the generation effect, characterized by the injection of additional features into the original image, effectively changing its class. Therefore, it is more meaningful to look at adversarial examples instead of adversarial perturbations. To enhance visualization, we proportionally scale the perturbations up and set their maximum value to 0.5 before adding them to the image.

B.3 Calibrating Output Values

Added Gaussian noise to the input image can alter the model's output, which in turn affects the generated adversarial perturbations. For instance, with the DeepFool attack, noise may cause the model to incorrectly classify the input image, thus preventing the algorithm from starting. Furthermore, certain algorithms target the class with the second-highest score. When noise is added, it can shift these score rankings, resulting in a change of the targeted class. To mitigate the effect of noise on perturbations, we have developed a method to calibrate the output values.

Our method first computes a calibration vector $calib.$, obtained by subtracting the output vector of the noise-augmented image $f(x + N(0, \sigma^2))$ from that of the original image $f(x)$. Then, while calculating the output values of models to generate perturbations $\delta(x)$, we add this calibration vector to counteract the effects of noise. The mathematical form is expressed in Eqn. 3. This approximation is justified by the local linearity property of neural networks (Goodfellow et al., 2015).

$$\begin{aligned} &f'(x + N(0, \sigma^2) + \delta(x)) = f(x + N(0, \sigma^2) + \delta(x)) + calib. \\ &calib. = f(x) - f(x + N(0, \sigma^2)) \end{aligned} \tag{3}$$

B.4 The Effect of Clipping

In the experiment, we do not clip adversarial examples to a range between 0 and 1. The reason for this is that clipping enhances the recognizability of perturbations by making them resemble the original image, especially in untargeted attacks. We aim to ensure that any observed resemblance arises directly from the perturbations themselves, rather than from the method of displaying perturbations.

We demonstrate the idea as follows: If the original image has a pixel value of 1, then after clipping, the corresponding perturbation will have a value of zero or less. This ensures that the resulting adversarial example will not have a pixel value exceeding 1. When we later display the perturbation, we multiply it by -1 and scale up its value to counter the negative sign brought by the masking effect, as described in Appendix B.2. Thus if a pixel value is large, clipping ensures that the displayed perturbations will also have a large value for that pixel.

In Figure 10, the observed clipping effect is illustrated. The clipped perturbation displays brighter colors and its contours align more closely with that of the original image, compared to the unclipped version. Note that in this example, clipping is applied to each individual perturbation before averaging.

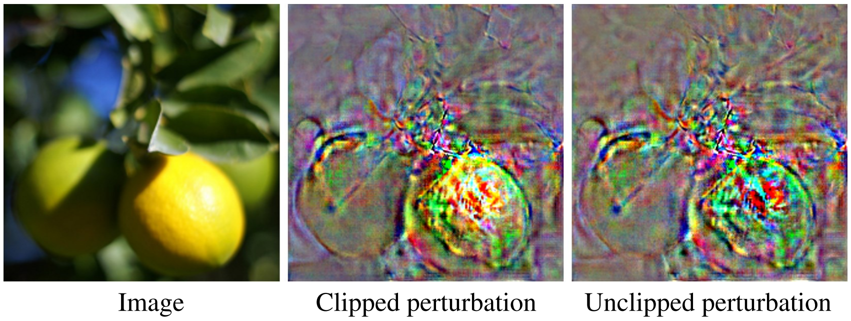

Figure 10: Clipping Effect for perturbations in MM+G Setting with BIM Attack. The panel shows a lemon and its clipped and unclipped perturbations. While the clipped version has colors and contours that are closer to the original image, this resemblance is artificially induced by the clipping operation.

C Experiments on MNIST and CIFAR-10 Datasets

C.1 Experimental setup

In this section, we outline the experimental setup for the MNIST and CIFAR-10 datasets, highlighting how they differ from our experiments on the ImageNet dataset. While the overall experimental framework remains consistent across all datasets, there are dataset-specific variations in the methods for data selection, the employed models, and the parameters used. This information will be detailed in the following.

C.1.1 Data Selection

For CIFAR-10 and MNIST datasets, we selected the first 10 images from each of the 10 classes in the testing sets. This resulted in 200 distinct images. For these chosen images, we generate adversarial perturbations under both SM and MM+G settings.

C.1.2 Model Architecture

For the MNIST experiment, we used 101 self-trained models, all based on the same VGG architecture. These models differ only in their initialization value. Of these, 100 serve as source models while the remaining one serves as a testing model. In contrast, for the CIFAR-10 experiment, we employed the entire suite of models available on the PyTorchCV repository (Sémery, 2018), totaling 70 distinct models. We selected four distinct architectures—DenseNet-40, DIA-ResNet-164, PyramidNet-110, and ResNet-56—as our testing models. The remaining 66 models function as source models. It is worth noting that under the SM setting, we specifically chose ResNet-56 in the testing models as the source model.

C.1.3 Experimental Parameters

For the MNIST experiment, the BIM attack is configured with parameters $\epsilon = 0.2$, $\alpha = 0.008$, and the number of iterations is set to 50. In contrast, the CIFAR-10 experiment has different parameter values for the BIM attack: $\epsilon = 0.03$, $\alpha = 0.0012$, and the number of iterations remains the same. For both MNIST and CIFAR-10 experiments, the CW attack parameters are consistently set with $c = 0$, $\kappa = 0$, and a total of 1000 iterations. For the DeepFool attack, η is 0.02, and the number of iterations equals 50 in both experiments.

For all experiments in the MM+G setting, the incorporated noise is sampled from an isotropic Gaussian distribution with a mean of 0. For the MNIST experiment, the Gaussian standard deviation is 0.2, and each image is copied 20 times, each incorporated with a different Gaussian noise, for every model. For the CIFAR-10 experiment, the standard deviation varies with attack algorithms. For both BIM and CW attacks, the standard deviation of noise is set at 0.05, and each image is copied 100 times with different Gaussian noises. Due to the time-consuming nature of the DeepFool attack, we reduced the number of copies to 20. Additionally, our empirical observations lead us to increase the standard deviation of Gaussian noise to 0.1 for DeepFool attacks on the CIFAR-10 experiment to further enhance the visibility of the generated adversarial perturbations.

C.2 EXPERIMENTAL RESULTS

Figure 11 demonstrates adversarial perturbations generated on CIFAR-10 and MNIST datasets under MM+G and SM settings. It is evident that under the SM setting for the MNIST dataset, the perturbations already exhibit distinct features recognizable by humans. These features become even more pronounced with reduced background noise under the MM+G setting, except for the CW attack where the SM setting already produces low noise and clear human-identifiable features. Interestingly, the MM+G setting also generates new, digit-like shapes in the perturbations, such as a blurred ”4” for the digit ”1” in the BIM attack and a clear ”7” in the CW attack. This is the generation effect. Further examples of the generation effect are evident during targeted attacks.

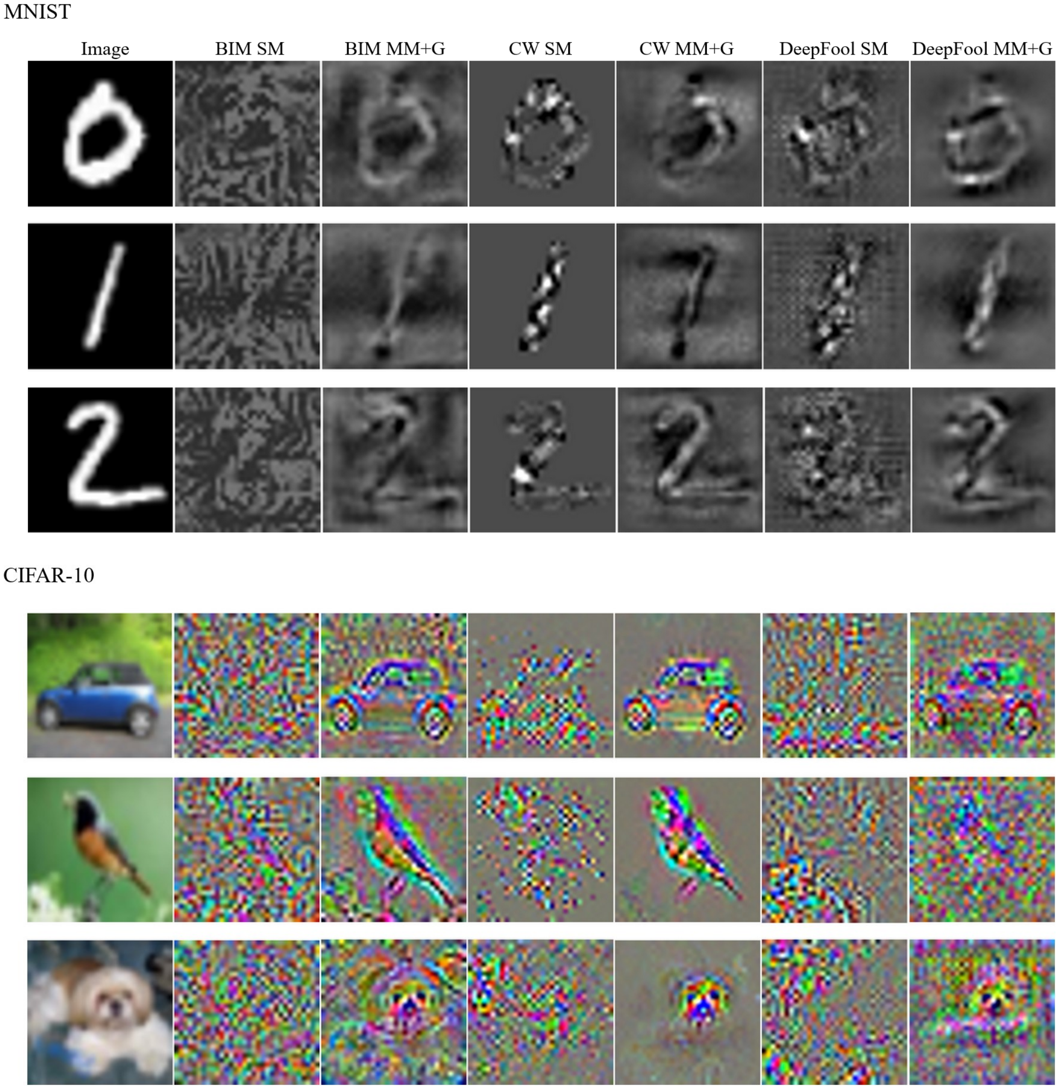

Figure 11: Adversarial perturbations generated under the MM+G setting for MNIST and CIFAR-10 datasets.

For the CIFAR-10 dataset under the MM+G setting, the perturbations for car and bird images exhibit identifiable features resembling the shapes of the input images. In contrast, the perturbations for the dog image primarily capture facial features. This suggests that human-identifiable features do not necessarily encompass an object's entire contour. While we present only three examples per dataset, it is important to note that the MM+G-derived perturbations consistently exhibit human-identifiable features. This consistency is observed across 200 images used in the experiment and mirrors the masking effect seen in our ImageNet experiments.

C.3 Quantitative Analysis of Experimental Results

To assess the recognizability of the generated perturbations, we conducted a machine evaluation test. Specifically, we checked if the model's predictions for each scaled perturbation aligned with the label of its corresponding image. This method of evaluation was applied to all the generated perturbations. It is worth noting that our evaluation approach is analogous to the one used in the ImageNet experiment, as detailed in Section 5.2.1.

In the MNIST dataset, according to the testing model, 57%, 44%, and 68% of perturbations generated by BIM, CW, and DeepFool attacks under the MM+G setting are correctly classified, respectively. Comparatively, BIM, CW, and DeepFool attacks in the SM setting have accuracy of 18%, 46%, and 17%, respectively.

In the CIFAR-10 dataset, we averaged accuracy across four testing models. These models can classify MM+G-generated perturbations with 63%, 46%, and 21% accuracy using BIM, CW, and DeepFool attack methods, respectively. The classification accuracy for SM-generated perturbations is significantly lower, yielding accuracy nearly equal to random guessing at 9%, 10%, and 10%.

The strong performance of testing models over random guessing indicates that the perturbations contain features essential for accurately classifying the original images, which is described as the masking effect earlier. Additionally, the non-trivial accuracy observed in the MNIST dataset under the SM setting suggests that these perturbations already include observable identifiable features, in agreement with the observation in Figure 11.

D More Experimental Data

In Figure 12 and Figure 13, we supplement our study with five additional input images, each accompanied by adversarial perturbations from five different attack algorithms, under untargeted attack mode with both SM and MM+G settings. To offer a general overview of the dataset, we display the first image and its corresponding perturbations from each class, according to lexical file name order in the ImageNet dataset. Figure 12 and Figure 13 show perturbations from gradient-based attack algorithms and search-based attack algorithms, respectively. Consistent with our earlier findings, we observe a pronounced masking effect for perturbations generated under the MM+G setting.

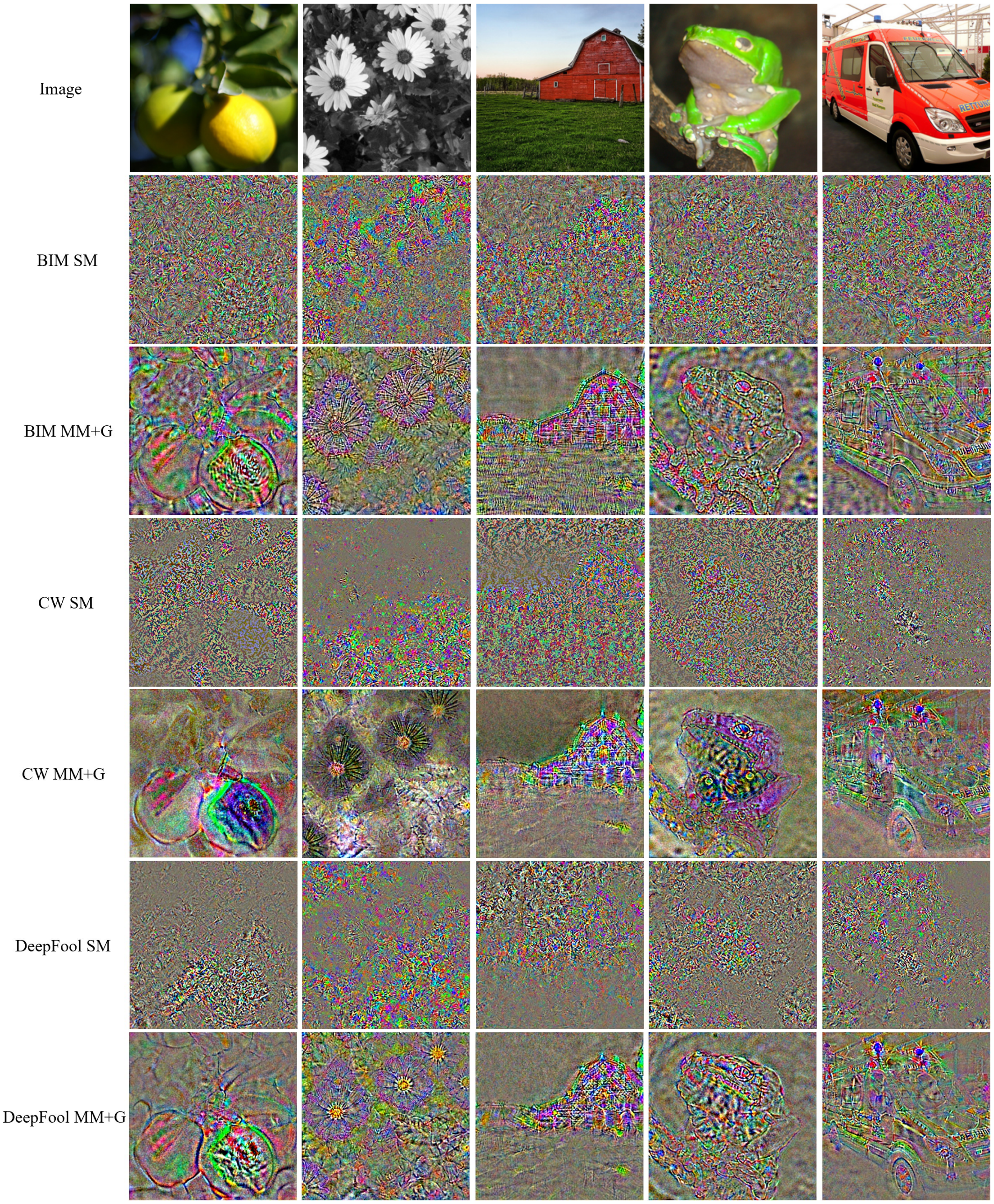

Figure 12: Additional examples of perturbations generated from gradient-based attacks.

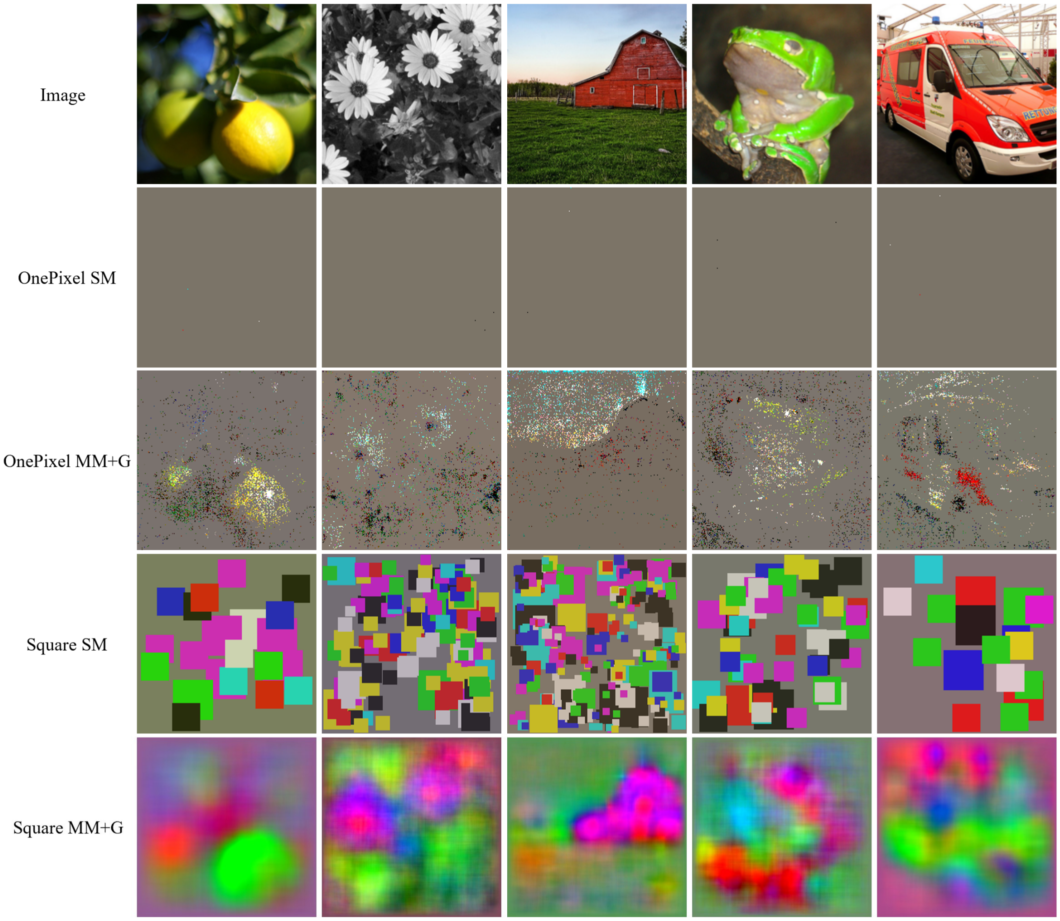

Figure 13: Additional examples of perturbations generated from search-based attacks.

E Comprehensive Information on Attack Accuracy

The attack strengths of the adversarial perturbations obtained from different attack algorithms for ImageNet are listed in Table 1. The Image column of Table 1 shows that, on average, the testing models correctly classify 81.8% of the input images. The effect of adding noise (Noise column), defined as random sampling where each pixel value has an equal probability of being +0.02 or -0.02, to images has a minimal effect on testing models' classification. The incorporation of perturbations in the SM setting lowers the classification accuracy to 63.3% for the BIM attack. In the MM+G setting, perturbations further reduce the classification accuracy to 13.2% for the BIM attack. This confirms the strong attack ability of perturbations generated in the MM+G setting.

Table 1: Attack strength of adversarial perturbations processed by MM+G.

Testing Models	Reference		SM			MM+G		
	Image	Noise	BIM	CW	DF	BIM	CW	DF
BN-Inception	81.5%	83.0%	64.0%	77.0%	68.0%	16.5%	22.0%	15.0%
DenseNet121	83.5%	83.5%	58.5%	77.5%	66.5%	10.5%	16.5%	13.0%
VGG-16	79.0%	79.5%	67.5%	76.5%	70.5%	12.5%	20.5%	17.5%
ResNet50	83.0%	82.0%	0.0%	7.0%	4.5%	13.0%	18.5%	14.0%
Avg.	81.8%	82.0%	63.3%	77.0%	68.3%	13.2%	19.7%	15.2%

F Contour Extraction

Figure 14 compares the perturbations generated by the BIM attack in the MM+G setting and contour-extracted perturbations via pixel-level annotation of the image of a daisy. The extraction of contours leaves a homogeneous background, where the value is set to zero, in the perturbations.

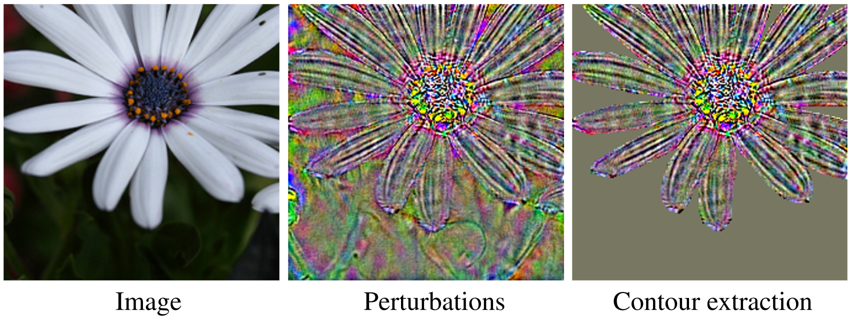

Figure 14: Visualizing contour extraction for a perturbation. From left to right, the input image of a daisy, perturbations generated by the BIM attack in the MM+G setting, and contour-extracted perturbations via pixel-level annotation.

G Parameters for Search-based Attacks

G.1 Square Attack

In the experiment, we chose the L_{inf} norm-based Square attack algorithm. We modified the initial perturbation value, switching it from the original stripe pattern to zero. This adjustment was made to focus our study on the human-identifiable features that emerged during the perturbation optimization process, rather than the effects that originated from the stripe pattern. The attack parameters are specified as follows: The perturbation magnitude is assigned a value of $\epsilon = 0.05$, and the variable representing the size percentage, p, is set to 0.1. Under the MM+G setting, we conducted 100 attacks for each image utilized by a model.

To generate adversarial perturbations of a single image, we need to average approximately 27,000 perturbations. This means that generating a complete perturbation using an RTX 3090 graphics card would take about 5 hours. Due to limited computational resources, we generated adversarial perturbations for 40 images, comprising 2 images from each of the 20 classes used in the experiment.

G.2 One-pixel Attack

For each attack, we executed 400 iterations using a population size of 200, targeting three pixels per perturbation. We performed 10 repeated attacks for each model with different initializations, averaging the generated perturbations for each input image. Due to the high computational demands, which require an average of 80 hours to generate a single perturbation in the MM+G setting on a NVIDIA Tesla V100 GPU, we restricted our MM+G experiments to 10 images, each from a different class[2].

H More Experimental Results on Targeted Attack

In Figure 15, there are three input images labeled as Siamese cat, slug, and lemon with the targeted classes being tiger, snail, and orange, respectively.

Under the MM+G setting, Figure 15(a), demonstrates a change in the cat's eye color from blue to orange and the emergence of black stripes that are features synonymous with a tiger. For Figure

[2] Selected classes: ambulance, barn, baseball, daisy, green mamba, lemon, teapot, tree frog, great white shark, and cock.

15(b), the perturbations tend to transform the curled slugs into the shells of a snail. Meanwhile, in Figure 15(c), perturbations change the lemons' color from yellow to a more orange hue, making the adversarial examples resemble oranges.

The above finding holds for both CW and BIM attacks. However, under the SM setting, it is challenging to observe the generation effect for targeted attacks. The observations are consistent with those from Section 6.

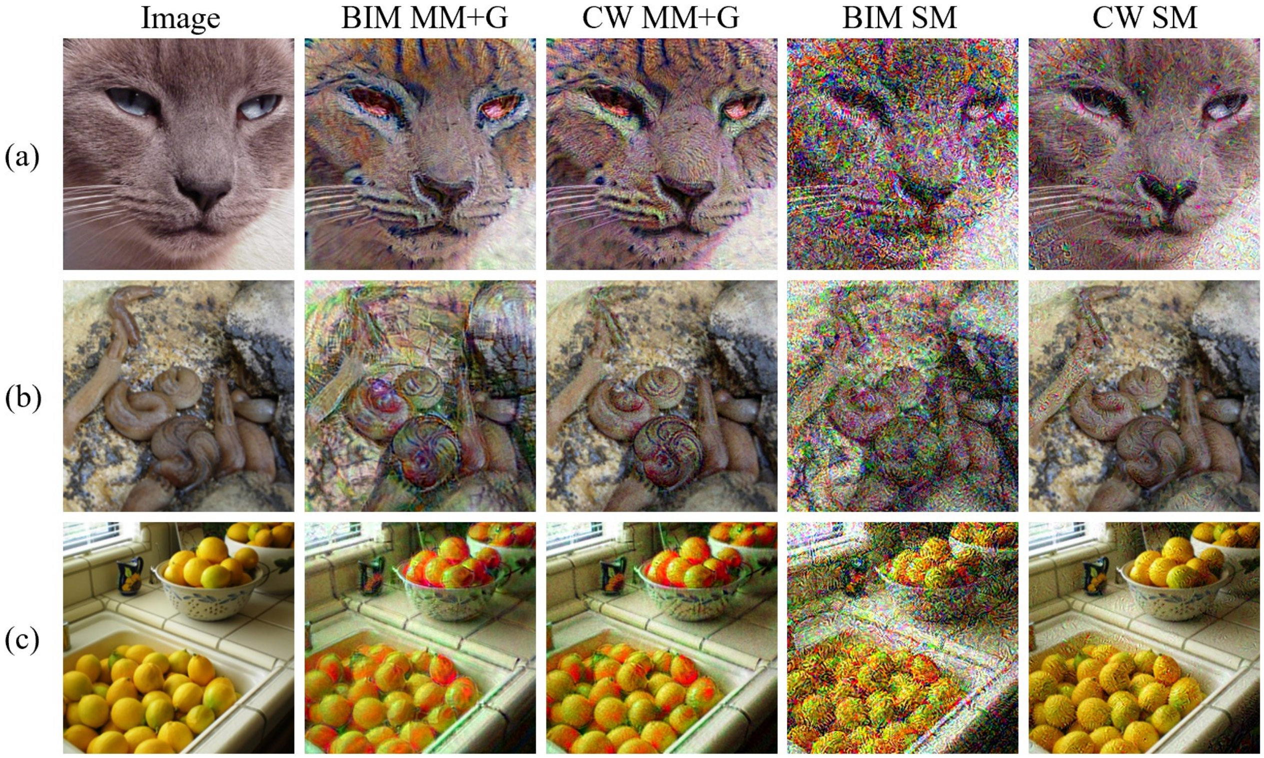

Figure 15: More examples of adversarial examples generated from targeted attack algorithm under MM+G and SM settings: (a) Transforming a Siamese cat into a tiger. (b) Transforming slugs into snails. (c) Transforming lemmons into oranges.

I Model Count and the MSE Convergence for Perturbations

This experiment investigates how the quantity of models employed for averaging perturbations influences the calculated mean square error (MSE) derived from 270 models in the MM setting. Our goal is to determine the number of models needed for the convergence of the MSE score, which is highly related to the emergence of human-identifiable features, and to understand how each model individually contributes to this MSE score.

Prior to calculating the MSE score, the averaged perturbations will be normalized using standard deviations and means from ImageNet datasets. This normalization ensures that the resulting MSE is on a comparable scale to the MSE scores calculated from images sampled from the ImageNet dataset, which have input values ranging between 0 and 1.

Figure 16 illustrates the outcomes of BIM, CW, and DeepFool attacks, showing the average MSE values between perturbations generated using varying numbers of models and those using 270 models. The x-axis represents the quantity of models employed in generating perturbations, and the y-axis shows the respective MSE values.

Figure 16 reveals a similar trend for perturbations from BIM, CW, and DeepFool attacks. To achieve MSE convergence within 0.05, an average of 25 models are needed for the three attack algorithms. For an MSE of 0.02, 90 models are required, while an MSE of 0.01 necessitates 157 models. The findings offer an estimate of the number of models required to attain MSE convergence in the MM setting, enhancing our understanding of how many models, on average, are necessary for the emergence of human-identifiable features.

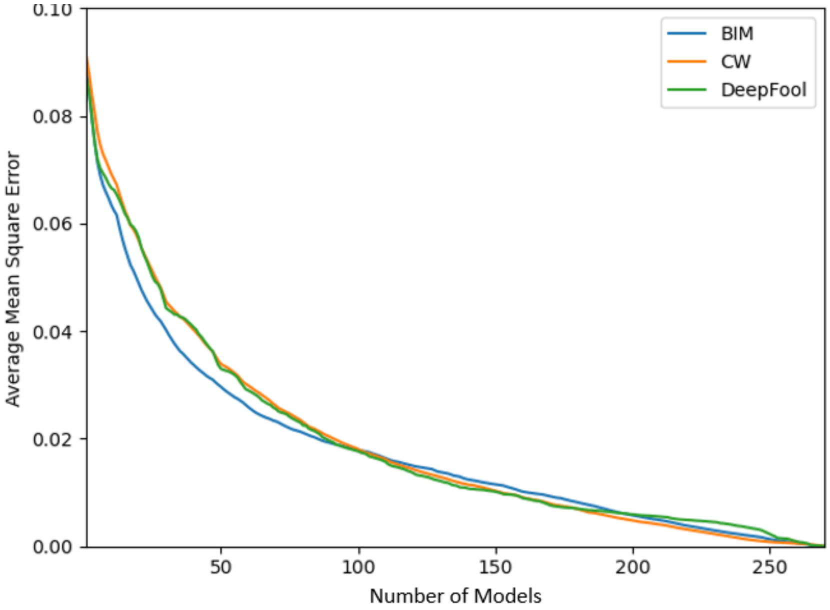

Figure 16: The averaged MSE scores between perturbations generated using varying numbers of models and those from the MM setting. The x-axis represents the quantity of models employed in averaging perturbations, and the y-axis shows the respective MSE scores.