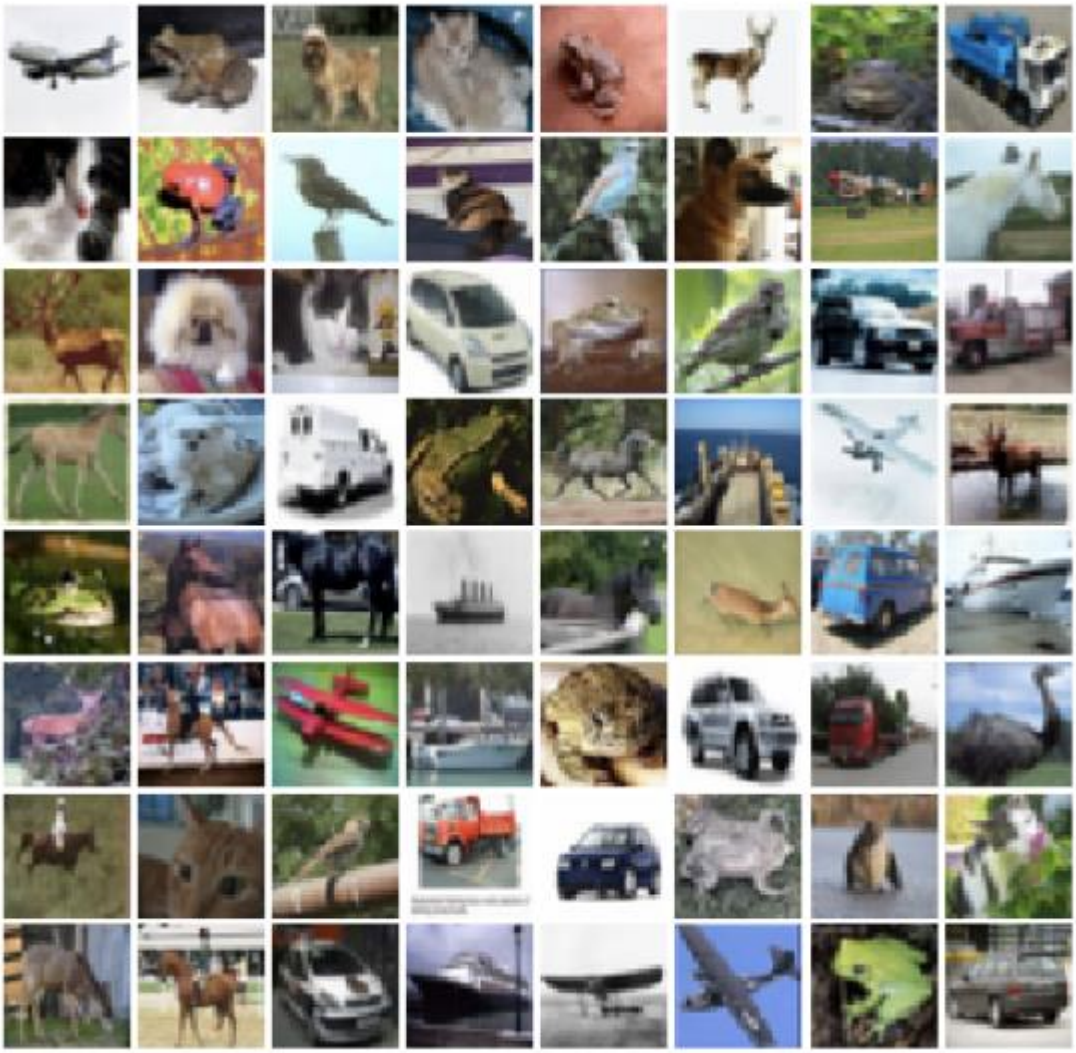
Report of 598 Homework 4

Bangguo Wang

Part 1

1. Perturb Real Images

Before the perturbation, the pictures of **X_batch** are as follows:



The classification performance by the **No_GAN** discriminator is:

```
[ True  True  True  True  True  True  True False False  True  True  True
  True False  True  True  True  True  True  True False  True  True  True
  True False False  True  True  True  True False  True  True  True  True
 False  True False  True  True  True  True  True  True  True  True False
 False  True  True False  True  True False False  True  True  True  True
  True  True  True  True]
```

Therefore, the accuracy is **0.78125**

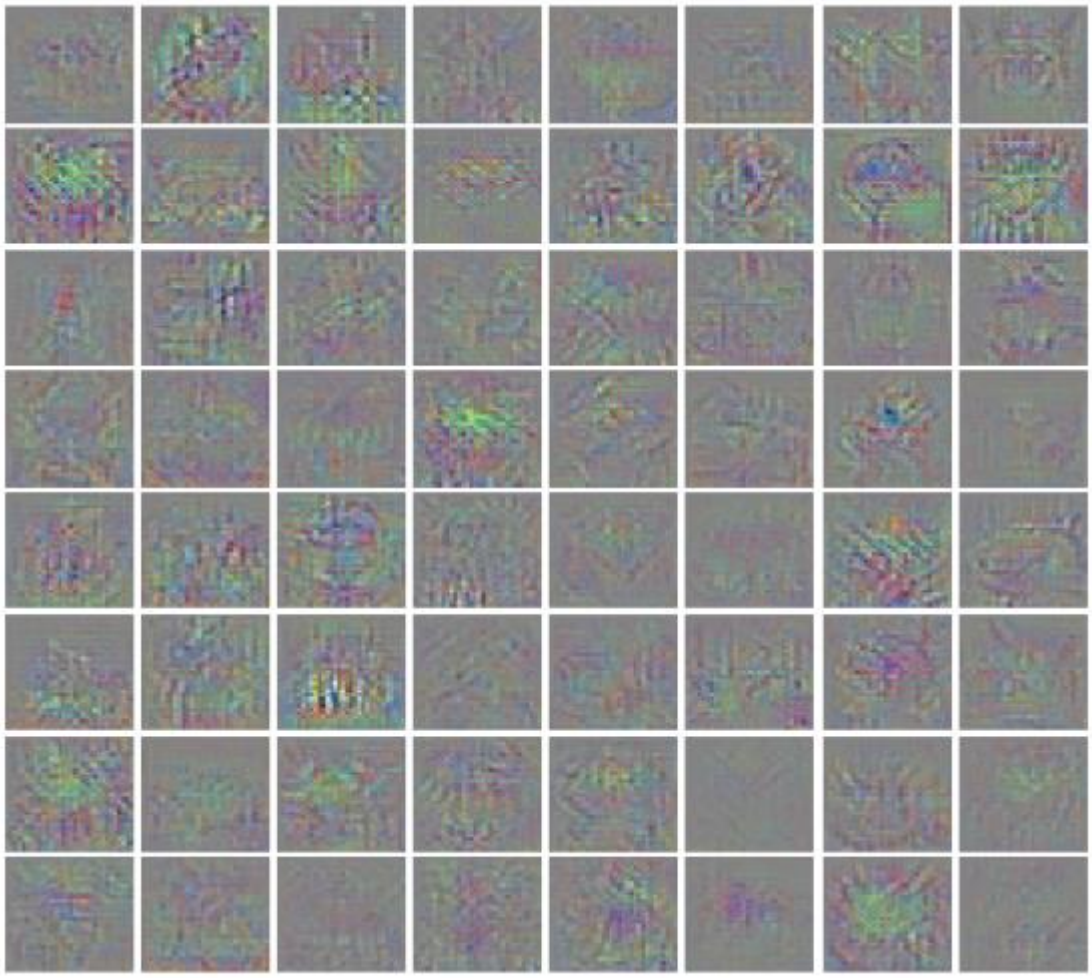
The classification performance by the **GAN** discriminator is:

```
[ True  True  True  True  True  True False  True  True  True  True False
  True False  True  True  True  True  True  True  True  True  True  True
  True False False  True  True  True  True False  True  True  True  True
 False False  True  True  True  True  True  True  True  True  True  True
 False  True  True  True  True  True False  True  True  True  True  True
  True  True  True  True]
```

Therefore, the accuracy is **0.84375**

Compared to the accuracy of **No_GAN** discriminator, the discriminator with **GAN** has higher accuracy performance.

Then we calculate the gradient based on the y_batch_alternate, the gradient images are as follows (**No_GAN** and **GAN** model has very similar images):



After the perturbation, the pictures of the **X_batch** are as follows:

which are very similar to the pictures before the perturbation.

However, the classification performance by the **Non_GAN** discriminator drops down greatly:

```
[ True False False False False False False False False False  True False
  False False False False False False False False False False  True False
  False False False  True False False False  True False False False False
  False False  True False False False False False False  True  True False
   True False False False False False False False False False False False
   True False  True False]
```

And the accuracy now is only **0.171875.**

As for the **GAN** discriminator, its performance is:

```
[False False False False False False False False False  True False
  True False False False  True  True  True False False False  True False
 False False False False False False False False False False False
 False False False False False False False False False False False
 False False False False  True False  True False False  True  True False
 False False False  True]
```

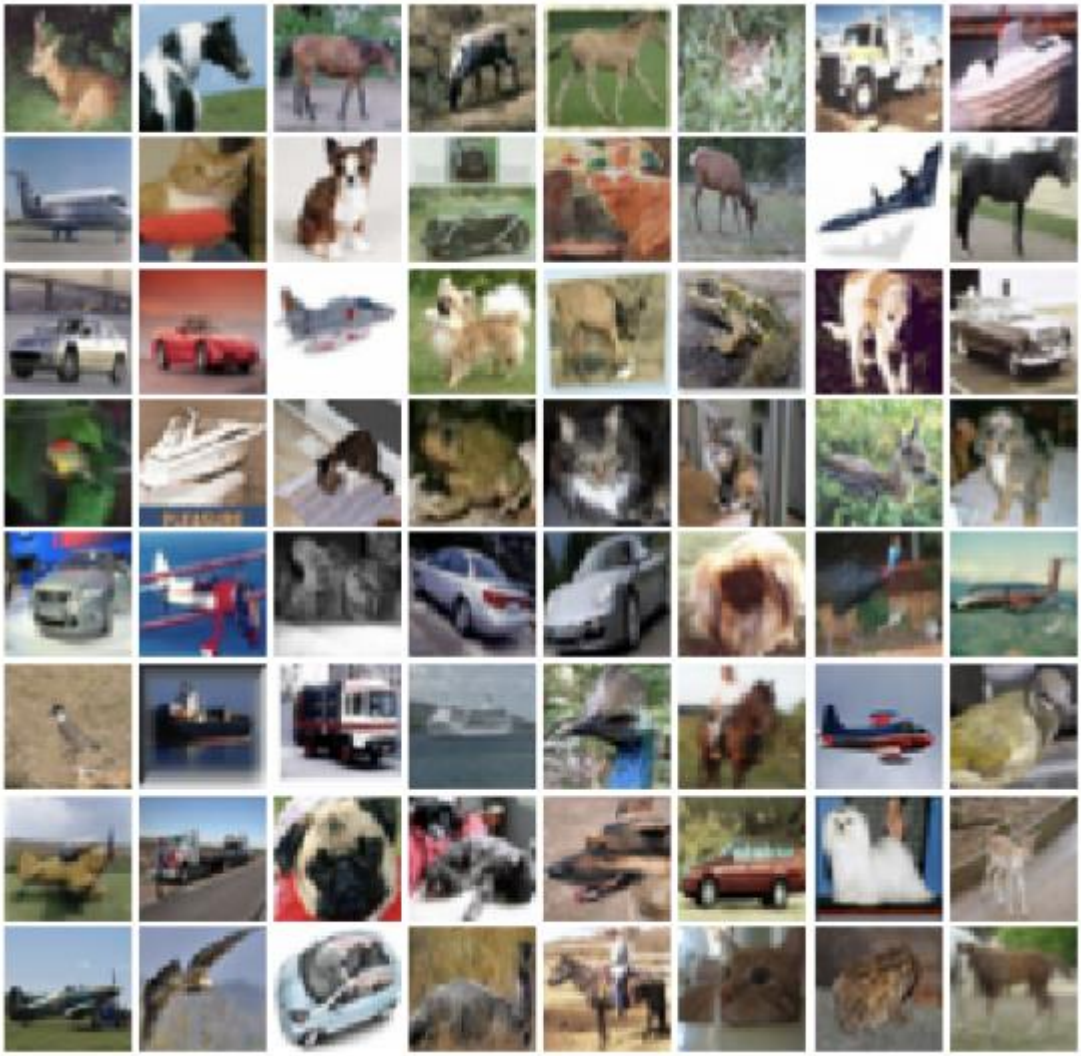
And the accuracy also drops down greatly, only **0.171875**

How it works:

Because we calculate the **gradient** based on the wrong labels **y_batch_alternate,** so when we use this gradient to modify the **X_batch** to be **X_batch_modified,** the **X_batch_modified** tend to go to the wrong direction that can reduce the loss based on the **y_batch_alternate** instend of based on the true label, which means **X_batch_modified** tend to be the labels of what the **y_batch_alternate** indicate. Therefore, when we use the discriminator to classify the **X_batch_alternate** and calculate the accuracy based on their true label, the accuracy must decrease a lot.

2. Perturb Image of Noise

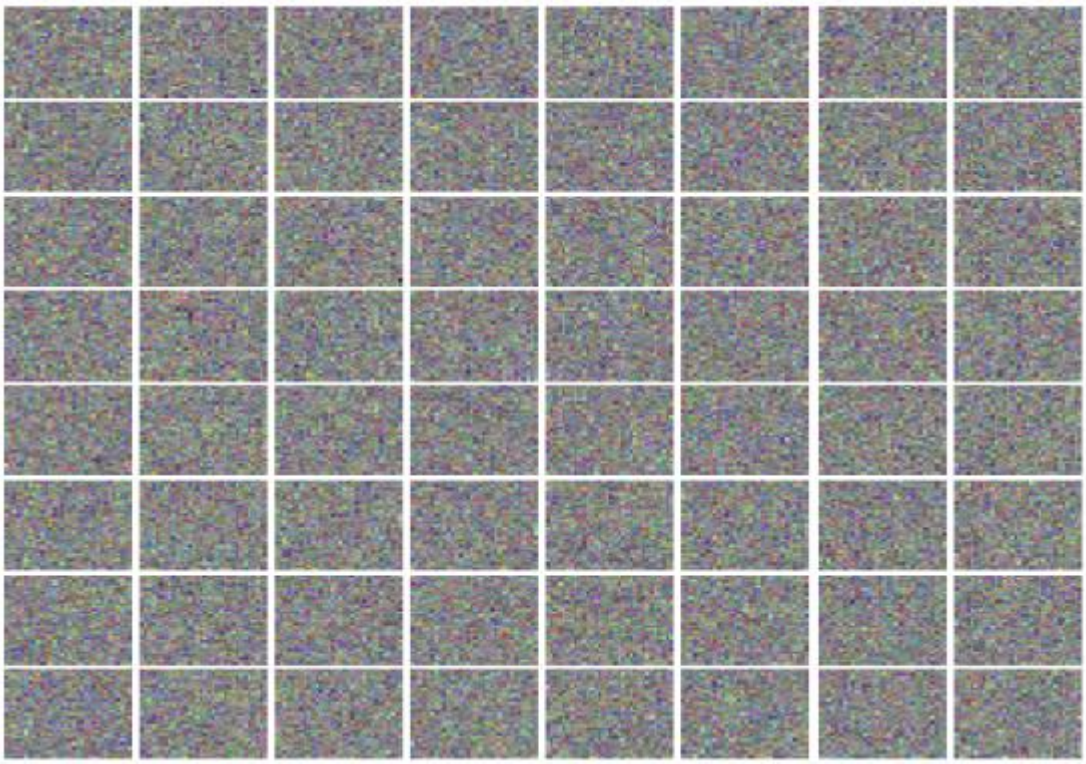
Before the perturbation, the original **X_batch** are as follows:



The accuracy by the **Non_GAN** discriminator is **0.8125**

The accuracy by the **GAN** discriminator is **0.859375**, much higher than acc uracy given by the**Non_GAN** discriminator.

The images of the noise are as follows:



after using these noises to perturb the images, now the **X_batch_alternate** images are as follow:

The accuracy for them by the **Non_GAN** discriminator is the same as it before: **0.8125**

The accuracy for them by the **GAN** discriminator slightly drops down compared to its performance before, which is **0.838125**

From the result, we can see that, the discriminator is robust, the perturbation of Gaussian random noise cannot have or have very little bad influence on its classification performance.

Besides, we can see that the discriminator with **GAN** training is better than that **No_GAN**, no matter before or after the noise perturbation.

3. High Activation Images for Output Layer

The high activation images are as follows (By **No_GAN** discriminator):

The high activation images are as follows (By **GAN** discriminator):

From the result, we can see that, using the **GAN** discriminator, the output layer has higher values in more pixels, so high activation images are clearer, which means the discriminator with **GAN** has higher ability of recognizing images.

How it works:

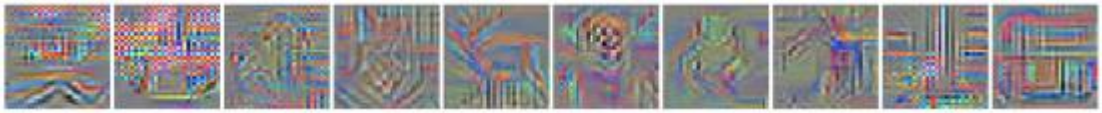
It calculates the partial derivative of the **output layer** with respect to **X_batch_modified**, then the **X_batch_modified** is updated by adding the **learning_rate*gradient** (with a **decay** term), which means

X_batch_modified is forced to be the one that can make the units of **output layer** have high values, which means **X_batch_modified** tend to be updated to be those images(features) that the discriminator can capture.

The following are the high activation images that generated with different **learning_rate** and **decay**:

1) **Learning_rate**: 0.6, **decay**: 0.005, **iteration**:500 (**No_GAN**)

2) **Learning_rate**: 0.6, **decay**: 0.005, **iteration**:500 (**GAN**)

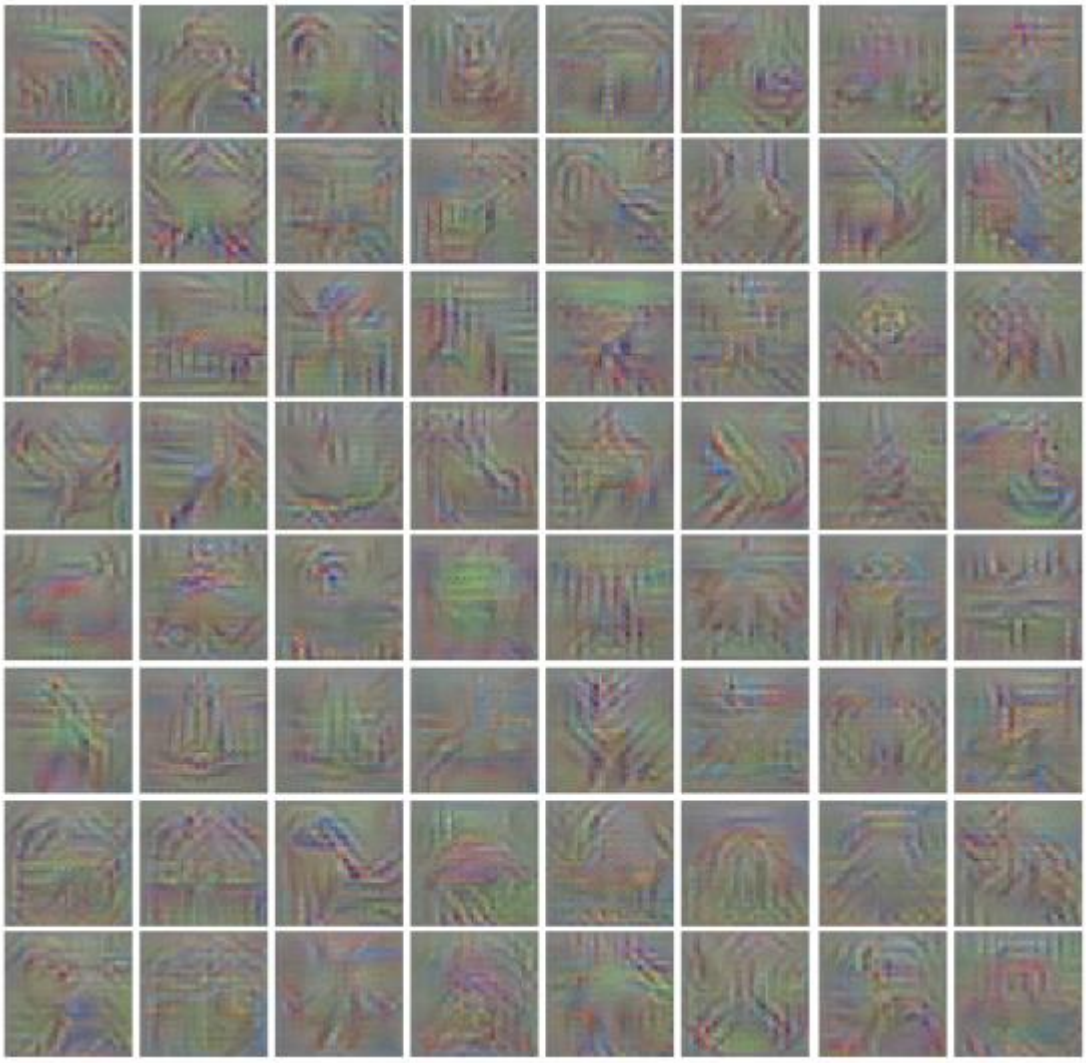


3) **Learning_rate**: 0.4, **decay**: 0.005, **iteration**:1000 (**No_GAN**)

4) **Learning_rate**: 0.4, **decay**: 0.005, **iteration**:1000 (**GAN**)

4. High Activation Image for Intermediate Features

The features images are as follows (**No_GAN**):

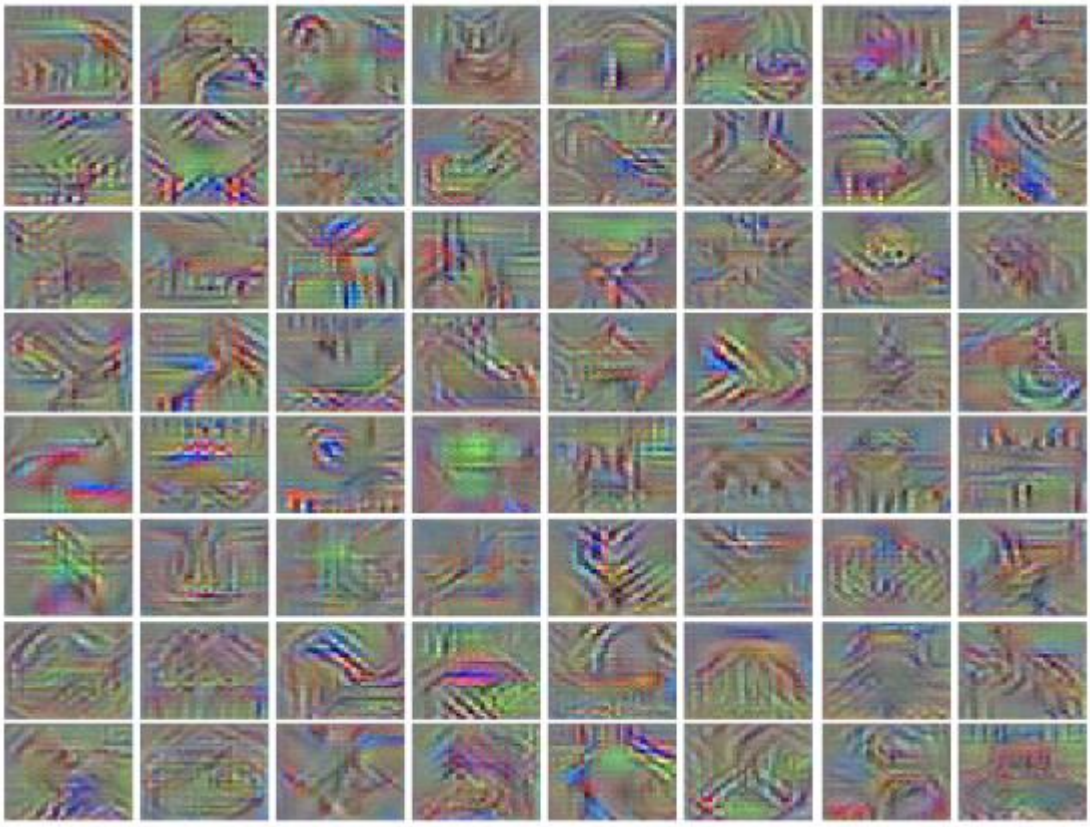


How it works:

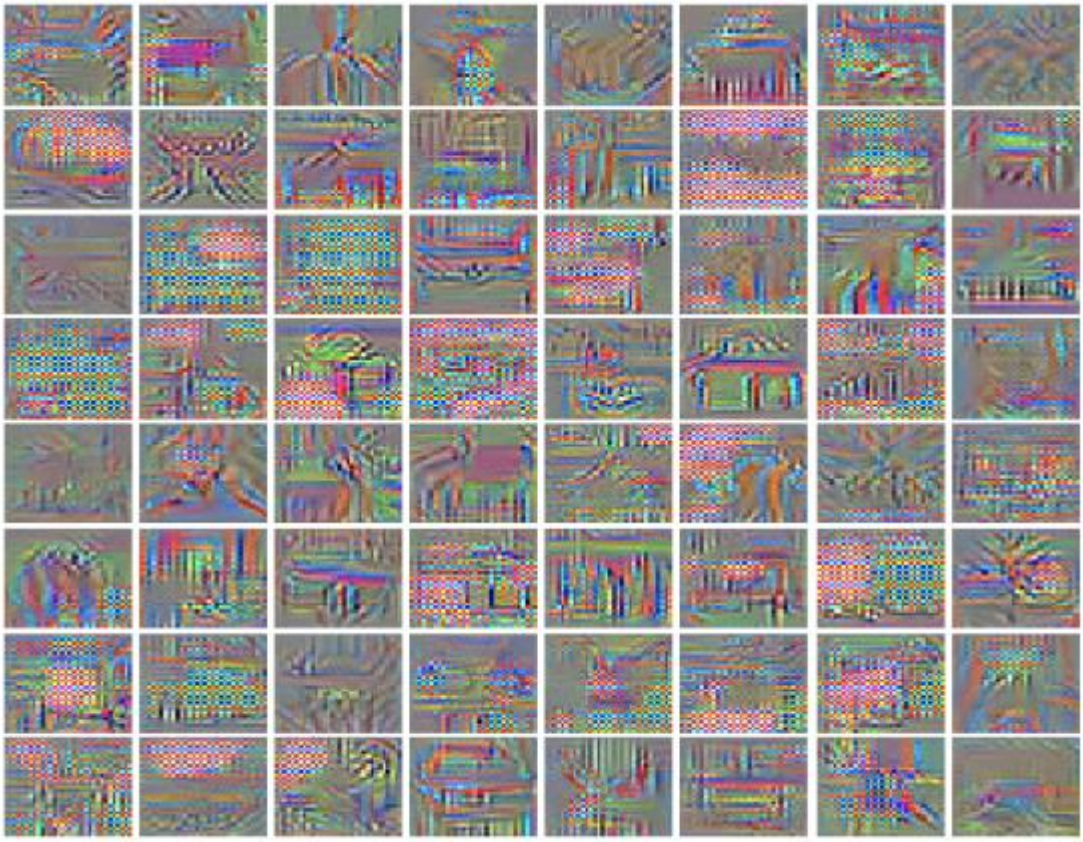
It calculates the partial derivative of the first 64 features of **the layer before the output layer** with respect to **X_batch_modified**, then the **X_batch_modified** is updated by adding the **learning_rate*gradient** (with a **decay** term), which means **X_batch_modified** is forced to be the one that can make the 64 intermediate features have high values, which means **X_batch_modified** tend to be updated to be those images(features) that the discriminator can capture.

The following are the high activation images that generated with different hyper-parameters:

1) **learning_rate : 2.0 ， decay: 0.002，iterations:1000 (No_GAN)**



2) **learning_rate : 2.0 ， decay: 0.001，iterations:1000 (GAN)**



Compared to the feature images generated from the **No_GAN,** the feature images by the **GAN** discriminator have higher value in more pixels, so the **GAN** discriminator can capture more details of the images, so it has higher confidence and ability of recognizing the images.

Part 2

During 250000 iterations, there are 500 images generated by the generator, the GIF of these 500 images can be found in this folder, named '**images.gif**'.

Besides, to see the change of the images more clearly, I also use 25 images (20 interval) to generate another GIF, which can also be found in this folder, named '**20_intervals_images.gif**'

From the result, we can see that the images become realer and realer as the iterations increase.