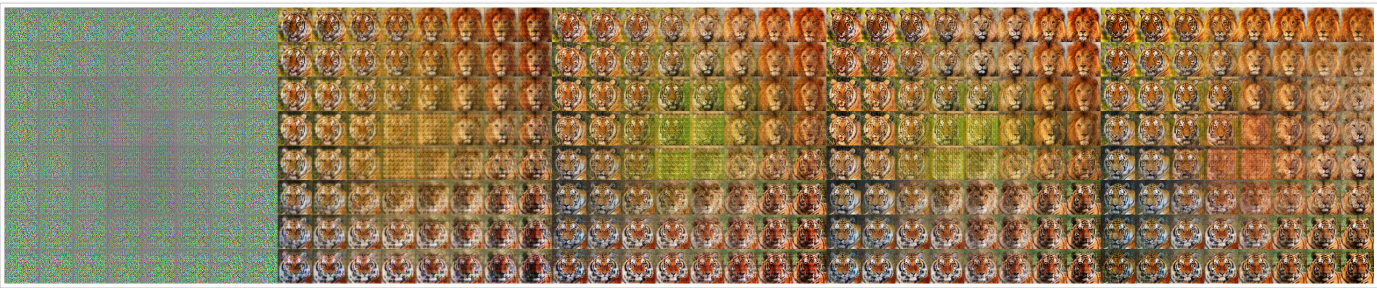
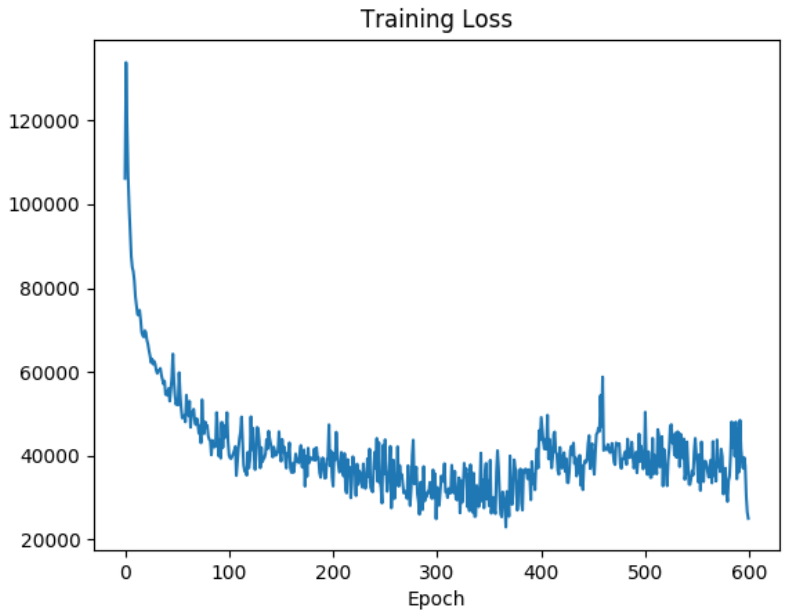
Generator Net

In [1]:
```
import os
# os.getcwd()
os.chdir('./GenNet')
%run main.py
```

```
/usr/local/lib/python3.5/dist-packages/h5py/__init__.py:36: FutureWarning: Conversion of the second argumen
t of issubdtype from `float` to `np.floating` is deprecated. In future, it will be treated as `np.float64 =
= np.dtype(float).type`.
  from ._conv import register_converters as _register_converters

Loading dataset: ./Image/lion_tiger
Data loaded, shape: (11, 64, 64, 3)
Start training ...
Epoch[ 0], train_loss: 106186.218750
Epoch[50], train_loss: 52933.664062
Epoch[100], train_loss: 40062.992188
Epoch[150], train_loss: 38601.976562
Epoch[200], train_loss: 32662.742188
Epoch[250], train_loss: 43796.933594
Epoch[300], train_loss: 32383.611328
Epoch[350], train_loss: 27960.781250
Epoch[400], train_loss: 49114.019531
Epoch[450], train_loss: 37361.585938
Epoch[500], train_loss: 50371.351562
Epoch[550], train_loss: 37657.484375
```





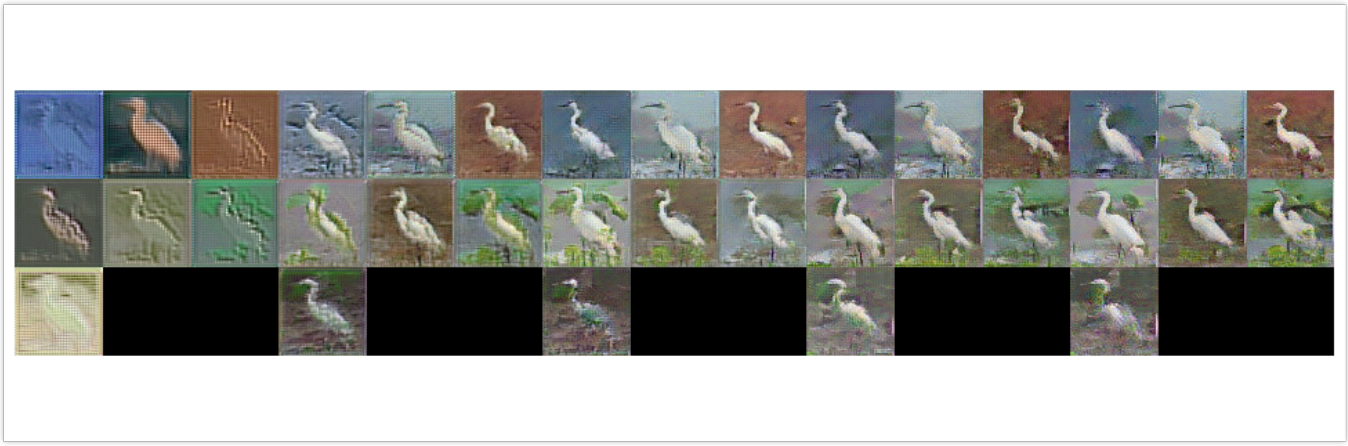
The figures above shows the training loss and generated images in 600 epoch. The training loss is fluctuating as we introduced randomness in the langevine sampling process. Despite fluctuations, the loss decrease monotonely， as the generated images increase in clarity. The model is a nonlinear generalization of factor analysis. It is proved by the interpolated images synthesized on a grid of two latent factors, from those images it can be observed that the two factors corresponds to the features of tiger and lion. On both factors, higher positive loadings gives rise to image resembling lion, while negative loadings represents the features of tiger.

Descriptor Net

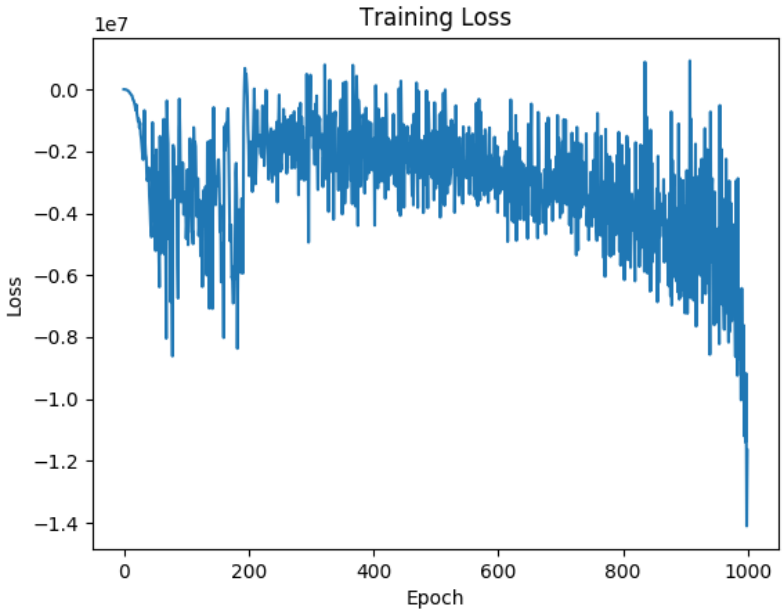
In [1]:
```
import os
# os.getcwd()
os.chdir('./DesNet')
%run main.py
```

```
/usr/local/lib/python3.5/dist-packages/h5py/__init__.py:36: FutureWarning: Conversion of the second argumen
t of issubdtype from `float` to `np.floating` is deprecated. In future, it will be treated as `np.float64 =
= np.dtype(float).type`.
  from ._conv import register_converters as _register_converters

Loading dataset: ./Image/egret
Data loaded, shape: (7, 64, 64, 3)
 Start training ...
Epoch[ 0], train_loss: 0.447842
Epoch[100], train_loss: -4821099.500000
Epoch[200], train_loss: -1786476.000000
Epoch[300], train_loss: -1484442.000000
Epoch[400], train_loss: -2397879.000000
Epoch[500], train_loss: -3194498.000000
Epoch[600], train_loss: -2350266.000000
Epoch[700], train_loss: -2374108.000000
Epoch[800], train_loss: -4010210.000000
Epoch[900], train_loss: -3783428.000000
```



The figure above showed the generated images in 900 epoaches. Because we use mean image as the initial input, the constucted image is colored patch at the beginning, then the quality of the images gradually increase in the training process as more complex features are learned.



The training loss for the descripter net is negative with big fluctuations. The output of the descripter net is a score for how "real" the input image is, so the training loss as the difference betweem the score for the fake and the real is negative. In the learning process, the descriptive ability of this score is also improved, so there are big fluctuations in the loss.