Semantic Understanding and Context Based Pixel Prediction

with

Semi-Supervised Learning and Generative Adversarial Networks

Sumanth Chennupalli

200851617

Queen Mary University London, UK

ec21083@qmul.ac.uk

Abstract

I present context-based pixel prediction. The images with random patches removed are given as input to Generator. The Generator will take it and then fill the patch and gives out the final output. These in-painted images will be given as input to the Discriminator which predicts that the given image is real or not.

The project includes trying out different large scale convolutional networks including Encoder Decoder architecture with bottleneck for training in a semi-supervised fashion. Then the trained model will be evaluated on multiple datasets to check the performance. In this project the context conditional GANs are used where Generator gives out the images by filling out the patches and Generator and Discriminator are conditioned on surrounding pixels. Semi-Supervised approach is used where we use both supervised and unsupervised by combing them where the unlabelled data is used to leverage the supervised task. I found output at initial epochs has lot of noise and then latter on context encoders learns the representation by optimisation which gave the accurate results. This project demonstrates the reconstruction of the image where the dark patch is placed.

1 Introduction

Humans have amazing ability to picture get the gist of the structure of the image after seeing a image even if there is a noise/missing pixels in between anywhere in the picture. When it comes to building a model that actually predicts the output of the missing patch of the images, the task becomes quite complicated and challenging. It’s natural ability that humans will understand the missing patch from in image that has a Tower Bridge of London or Eiffel Tower or any other famous monument for that matter and the part that has noise will can be filled by drawing the image in the imagination. but when it

`

comes to solving this problem using computer vision techniques it takes lot of training data and unique model architecture which gives out state of the art results. For example, if we see the Fig 1, there’s a missing part at the center of the image which makes it unclear to understand what the image is. But for humans, it’s quite a easy task to identify that it’s a Eiffel Tower.

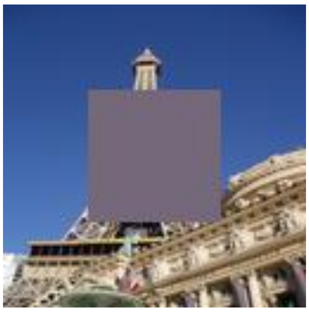
Fig 1

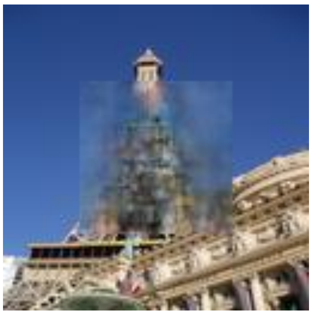
Fig 2

Here we build a context encoder which encoders and captures the context of the image which is similar to encoder and decoder architecture but here we use a bottleneck layer which has higher dimensions rather than lower dimensions which is generally the case while using encoder-decoder architecture. Since we are not trying to generate the same image, but we are trying to fill the missing image we don’t use low dimensional bottleneck layer. Also, using the high dimensional layer will try filling the missing patch as per pixels that are available nearby. This task needs much deeper semantic understanding of the image which makes this much more challenging than the normal image generation task.

We train the model in unsupervised method. At first, the model produces the random noise in the missing part of the image but however, once the model learns more deeper semantic understanding of the image the output makes much more sense at filling the missing part of the image. To achieve this, we basically have to minimize the loss we get, so we try to minimize both reconstruction loss which is also L2 and adversarial loss which helps to get the much more accurate results by getting the relation to the context of the image and taking up the mode for distribution. This method shows the great impact in filling the hole with semantic understanding of the context in the image.

2 Background

As we all know that there are multiple Computer Vision techniques that gives out excellent results in solving problems of image recognition, object detection and other video/image related problems. CNNs made a tremendous progress in the field of research on solving the image/video problems in both supervised and unsupervised manner.

Few of the state-of-the-art models including VGG, ImageNet, InceptionNet are trained on millions of images with labels which makes the models to learn the features from the image accordingly. But when it comes solving the problem of missing patch from the image, it has to be unsupervised since there won't be any labels and semantic information from the images has to be extracted only using the actual images. This problem resembles one of the Natural Language task which is basically to fill the missing words in the sentence by understanding the context.

Image generation is one of the main and significant part of this project and referring to new architectures such as Generative Adversarial Network which gave out really good results. In this project, the generator model is developed and trained using adversarial and reconstruction loss for generating images. Then a discriminator is used to discriminate the generated output and the goal is to fool the discriminator. This is another major AI technique which is used to solve this particular problem.

3 System Description

The system mainly consists of three main parts. Image processing is one of the core parts of the system design for this project where the image will be read using data loader and slice the image so that it will give the center part of the image. Then the main image will be cropped, and the dark mask will be implanted so that it'll the actual input for the Generator.

Secondly, the model architecture which is implemented using Generative Adversarial Network architecture plays main role where there are two models including generator and discriminator. The main goals revolve around Generator taking the input image with dark mask and the goal of the input is to predict the missing part of the image. Then the discriminator will be used to discriminate the original image and the image generated by the Generator. The goal here is to make sure that discriminator confuses to discriminate the generated image by Generator. Here we build the generator that has a bottleneck in the middle of the model architecture. This basically called encoder model which will encode the input image and try to rebuild it after processing through bottleneck. Here we use high dimensional bottleneck to encode the image, and this is basically to encode the image, the main reason for this is that since the goal is not to reconstruct the image but to decode the missing

`

part of the image using the semantic meaning in the entire picture. The input image with the missing patch will be given to the Generator then label this generated image as fake image and then this image will be process through Discriminator. Now using binary cross entropy loss, the loss will be obtained and then backpropagation follows. Now, the total error will be calculated by adding up error while discriminating the real image and the error while discriminating the fake image.

Here we use two losses, one is reconstruction loss (L2) which helps in minimizing the error in getting the semantic meaning of the image and another one is adversarial loss which basically deals with mode of distribution. GANs influenced a lot to use the adversarial loss to generate the image.

$$G\_L2 = (G\_output - Real\_image)^\wedge 2$$

As we already know, the L2 loss is calculated using mean square. Then the adversarial loss is calculated and minimized i.e., basically to maximise the log likelihood.

$$L(adv) = maxD(E[log(D(x)) + log(1 - D(F((1 - M)x)))$$

Then the generator will be optimised by backpropagation and this model will be trained for multiple epochs and the outputs at each epoch will stored for evaluation.

Coming to model, LeakyRelu is used throughout the Generator network for activation function and batch normalization is also applied throughout the network. After the bottleneck layer, the inverse conv layers to generate the image after feature extraction and Relu activation function is used till the last but one layer of the network and then Tanh is used as activation function as last layer. Discriminator model takes image as input and then we use LeakyRelu as activation till last but one layer and uses sigmoid at final layer to classify the real and generated image.

4 Dataset

I have decided to explore the city with various famous monuments, skyscrapers and historical architectural buildings and use the images of that city as dataset. Paris has some of the best monuments and tourist spots in the world. So, the dataset is prepared using various elements of the city Paris including Eiffel Tower, normal images of Paris streets, tourist photos which were taken in Paris, photos of some of magnificent places and historical architectures in Paris.

Few of image augmentation techniques has been used in this dataset and at the end the input image is normalized with mean and standard deviation.

5 Experiments and Results

Initially, the results aren't quite impressive, and the model is not able to capture the semantic meaning of the image. I have tried to use the complicated and huge architectures including AlexNet but due to heavy GPU requirements I wasn't able to train it and then using a bit simpler architecture the model is trained using the dataset of Paris.

During the initial epoch the model has produced noise and the loss of discriminator is at *~1.8* and the loss of generator is at *~8.3* and the L2 loss of *~1.06.*

`

Fig 3: Real ground truth images during first epoch

The images in the figure 3 are the ground truth of the dataset and they’re the ones that would be desired output of the model. We remove the part of the image in the center and keep the mask and gives it as a input to the model so that the model will try to fill the missing part based on the context.

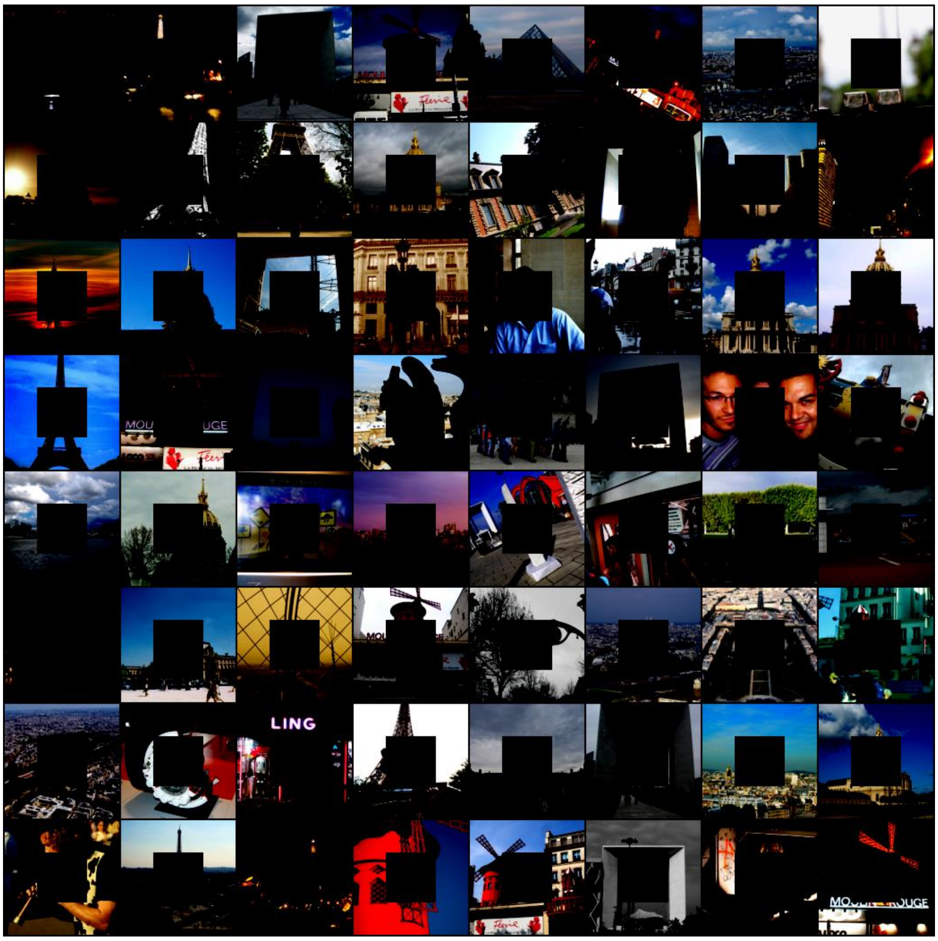

Fig 4: Input images to the model with the missing part in image

The images in figure 4 are the input images to the model with the missing part. The model will extract the semantic meaning of the image and understand what data would be filled in the missing part.

Now let’s look the predictions during the initial where model has not yet learned how to get the semantic meaning of the image to fill the missing part with meaningful pixels.

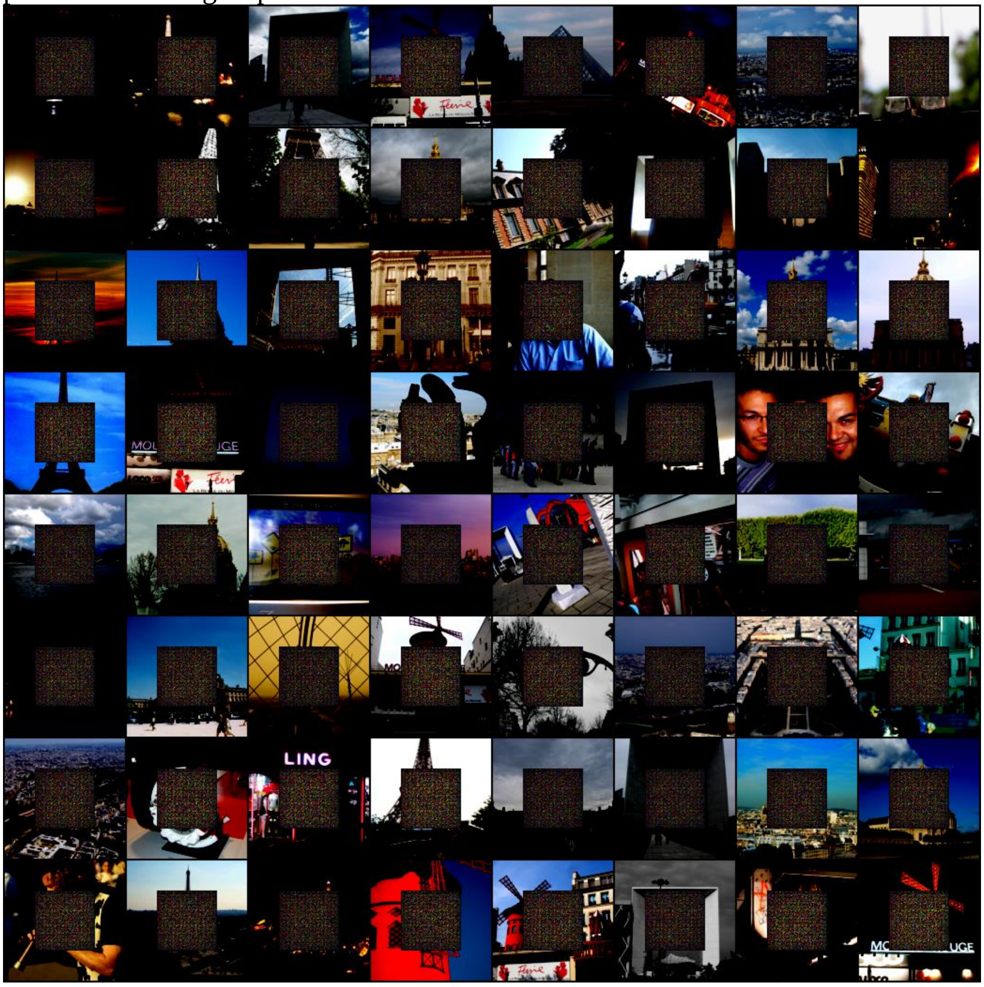

Fig 5: Generated images during the initial epoch

Now after making the model learn the feature extraction and semantic understanding of the image, it’ll be able to fill the missing part with data that makes more sense.

Let’s look at how the outputs are after training the model for 50 epochs. The expectation is that it has learned how to fill the missing part of the with more meaningful pixel.

`

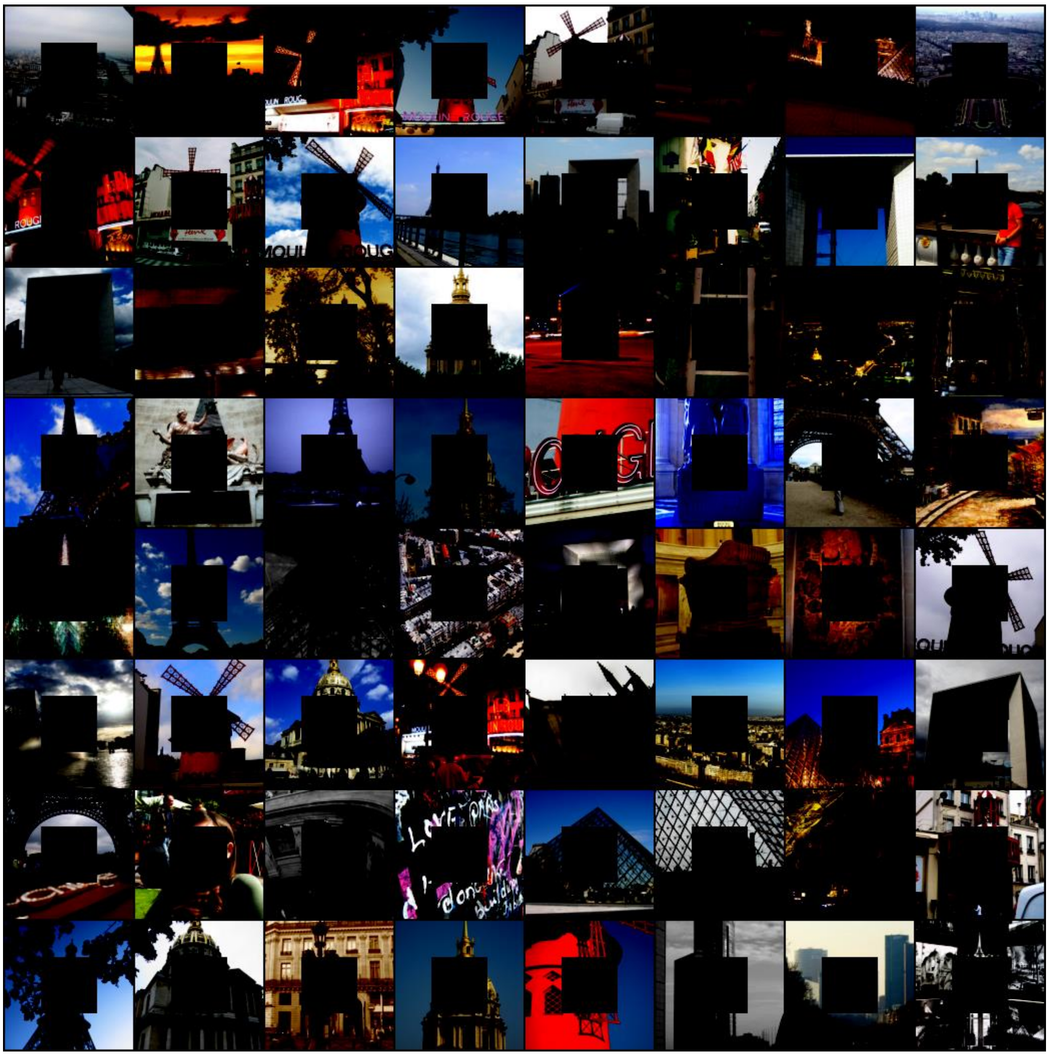

Fig 6: Inputs for 50th epoch

Figure 6 shows the input images to the model at the end of the 50th epoch of model training. Since the model has gone through significant amount of training the expectation is to predict the missing part of the image with meaningful pixels. Now, let’s see the output of the images after 50th epoch.

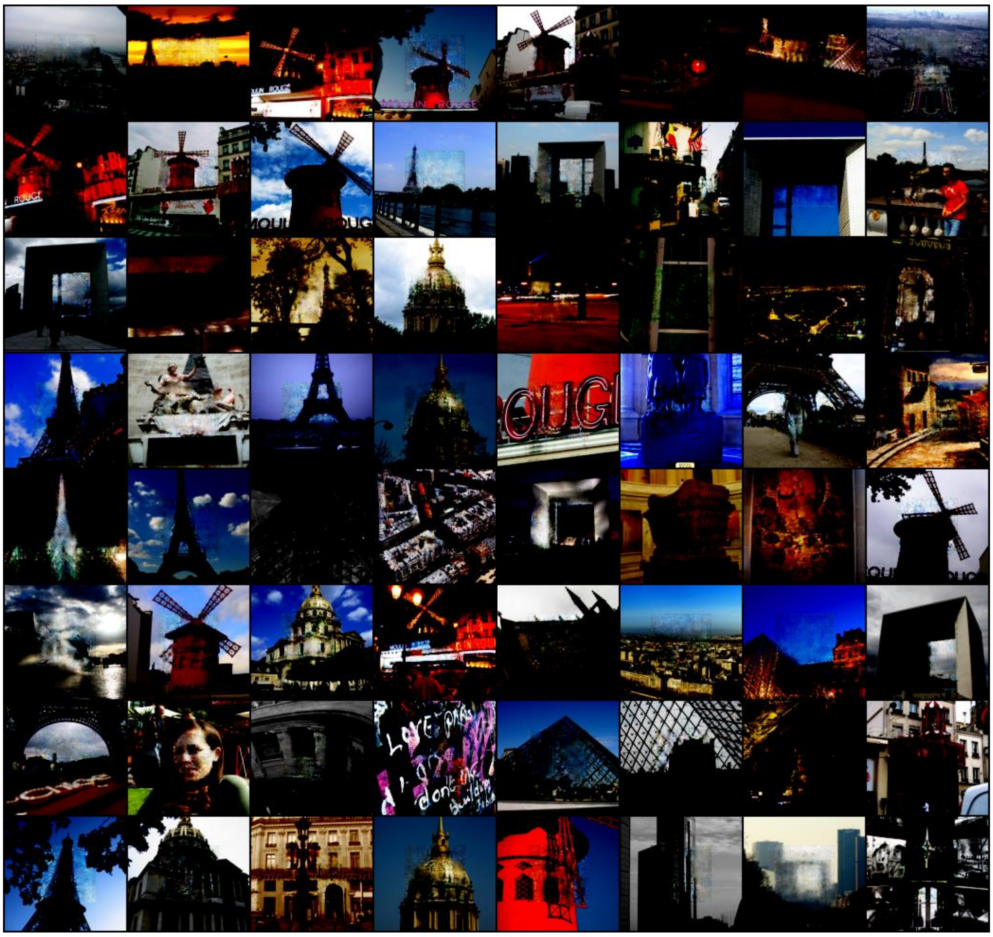

Fig 7: Generated images at 50th epoch

We can see that the missing part is filled with meaningful pixels which means the model has learned. Discriminator loss at this epoch is *~0.47*, generator loss is *~0.60* and the L2 loss is *~0.04*.

Model is trained over 200 epochs on GPU to get the state-of-the-art results and optimize the parameters. The model artifacts of both discriminator and generator will be stored for inference process. Now, let's look at the results after 200 epochs training.

`

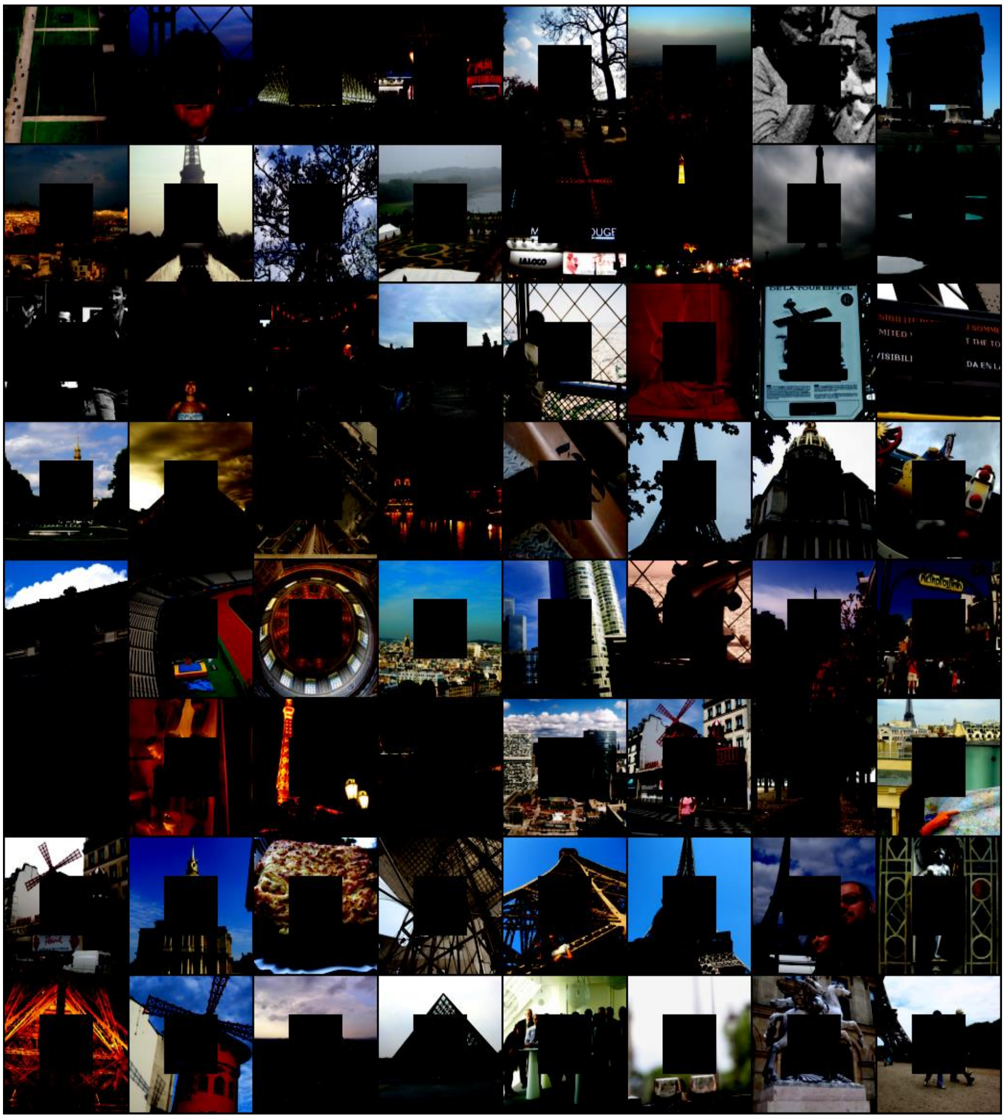

Fig 8: Input images at 200th epoch

Figure 8 shows the input images to the model at 200th epoch with the missing part of the image. Now that model has learned over 200 epochs, the expectation is to extract the features and fill the missing part.

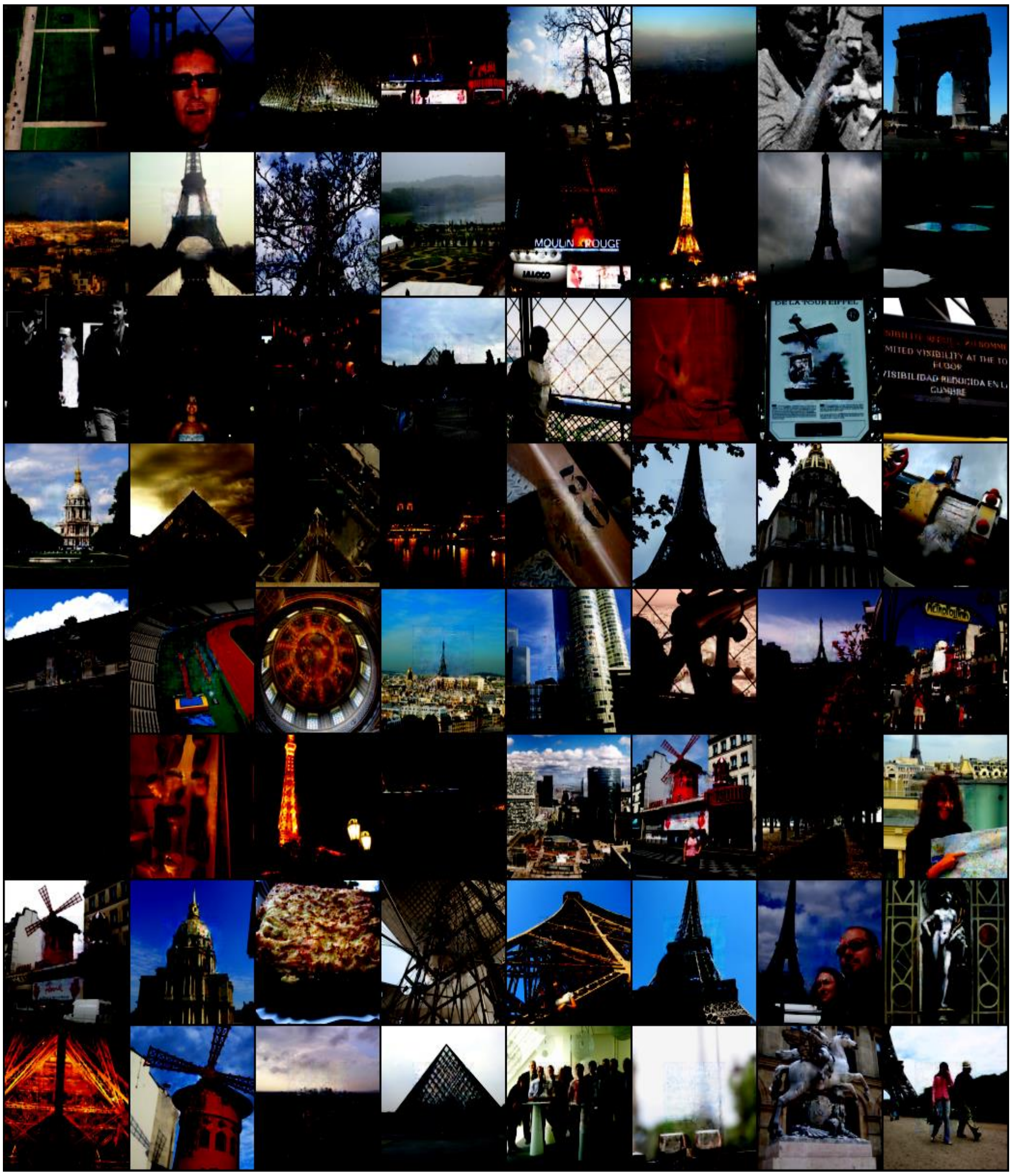

Fig 9: Generated images after 200 epochs

We can see the significant improvement in predicting the missing part and how semantic understanding got improved over training 200 epochs.

`

6 Discussion

The model has given out very intermediate results during the initial epochs of training and over the time the model is able to capture the features and able to get the semantic meaning of the image and fills the missing part with most apt pixels.

I would say that the model is able to capture the context in the image to fill the missing part, it may not be as good as the human imagination.

Epoch	D_loss	G_loss	L2_loss
50	0.4771	0.6042	0.0403
100	0.6980	3.6362	0.0276
150	0.5384	1.3744	0.0188
200	0. 4172	3. 9484	0. 0184

In the above table we can see the losses over epochs, the L2 loss decreased constantly over the epoch but it fluctuated a lot during the last 50-75 epochs. The surprising element is that how generator loss is varied over the training period, this would need further investigation.

7 Conclusions and Future Work

To conclude, this model will do the feature extraction of the image and get the semantic understanding on how to fill the missing part of the image using the encoder and decoder architecture with high dimensional bottleneck. This approach of inpainting the missing part of the image will work as a magic eraser and can be considered as a proof of concept for various innovative projects.

Future work includes getting better results without any noise in the output. The current implementation is good at getting the semantic meaning and filling the missing part to some extent, but it fails at filling the edges and lot of pixels to get the more clarity. It hard to fool the discriminator with these results and it can get lot better with the push this project has given.

Furthermore, this kind of projects will help highly in space research projects where we get high resolution images from various telescopes and merge them to get the final image. There are lot of situations where there is a crucial missing part in these images and this project will help filling it up with proper training. To add more, real time object detection with filling the missing part will be a game changer if it provides state-of-the-art results.

References

1. Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio Generative Adversarial Networks. *Statistics and Machine Learning(2014).*

2. Bastien, F., Lamblin, P., Pascanu, R., Bergstra, J., Goodfellow, I. J., Bergeron, A., Bouchard, N., and Bengio, Y. (2012). Theano: new features and speed improvements. Deep Learning and Unsupervised Feature Learning NIPS 2012 Workshop.

3. M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In ECCV, 2014

4. Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, Alexei A. Efros Context Encoders: Feature Learning by Inpainting. In *CVPR, 2016.*