ATTENTION-BASED DCGAN FOR IMAGE INPAINTING

CS282 (SPRING 2019) PROJECT REPORT

Remi Denoyer
Department of Mechanical Engineering
University of California Berkeley
Berkeley, CA 94720
remi.denoyer@berkeley.edu

Nicolas Hivon
Department of Civil Engineering
University of California Berkeley
Berkeley, CA 94720
nicolas.hivon@berkeley.edu

Clement Ruin
Department of Industrial Engineering
University of California Berkeley
Berkeley, CA 94720
clement_ruin@berkeley.edu

May 14, 2019

ABSTRACT

In this paper, we propose an Attention-based GAN architecture which allows both generation abilities and attention on context for image inpainting tasks. If previous works on GANs proposed good models to generate images from scratch, our architecture takes into account the surroundings of the missing part of an image to rebuild this missing part. Our generator first builds an approximate reconstruction with ApproxNet and then refines it using attention on the context of the image. Our final model achieves great results on Cars and Celebrity Faces datasets, with SSIM similarity scores above 0.95. The attention layers show a focus on relevant regions of the image regarding the masked part (neighbor pixels or similar shapes in the image).

Keywords Generative Adversarial Networks · Attention · CNN · Image processing

1 Problem Statement and Background

Image inpainting consists of reconstructing the missing parts of an image so that observers can not say that the regions have been restored. This technique is often used to remove undesired objects from an image or to restore damaged parts of old photos. If painting can be done by hand, deep learning researchers have developed algorithms for automatic painting. In addition to the image, it is also necessary to choose a mask to indicate the regions to paint.

Historically, image processing techniques have proven their potential to solve the problem of image inpainting. A famous approach called Exemplar-based-Image Inpainting was introduced in 2004 by Criminisi and al. [1] and used texture synthesis and pixels propagation to remove large objects from images.

More recently, the literature indicates that deep learning Generative Adversarial Networks models were powerful for this task. This model was first introduced by Goodfellow and al. in 2014 [2] and works with two networks: a discriminator and a generator. The first one aims at detecting fake images from real images while the second is trained to generate more and more realistic pictures from noise.

In this paper, we focus on the second approach, using GANs to eliminate damaged parts of pictures, and fill the background in accordance to the scene. In a similar fashion to Yu J. and al. [3], we add a attention layer to guide the generator though the background of images to reconstruct even more coherent and realistic images.

2 Approach

2.1 Dataset

We decide here to use consistent datasets, meaning that they are populated with similar images, to train and test the GAN. This choice is motivated by the network needing a shorter training with a consistent dataset.

The first dataset used is the Cars Dataset (Figure 1) created by Jonathan Krause and al [4] available at `http://ai.stanford.edu/~jkrause/cars/car_dataset.html`. This dataset is composed of roughly 16,000 RGB pictures of cars in different landscapes, positions, angle and under different lightning conditions.

We also test our implementation on the CelebFaces dataset (Figure 1) created by Ziwei Liu and al [5] available at `http://mmlab.ie.cuhk.edu.hk/projects/CelebA.html`, and compare the scores on both datasets.

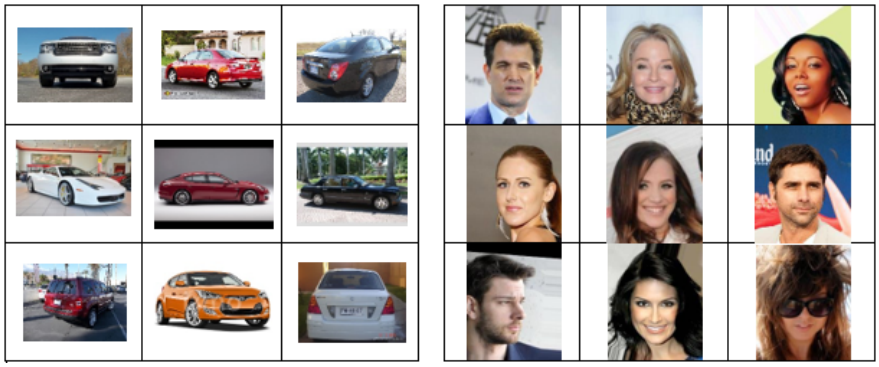

Figure 1: Examples of images in both datasets

2.2 Generating masked images

For both sets, we generate damaged images for our training set by adding random masks (with both random position and size) and defaults covering 2.4 to 10% of the original images (Figure 2 and Figure 3). Dimensions were reduced to 64*64 pixels in order to get a fixed image input size for all models. The test set is composed of the real undamaged pictures.

Entirely random mask Initially, we decide to create a mask with all components are random. We draw one upper left point coordinates, the mask's width and height randomly, between 10 and 20 pixels. Please note that the masks' shapes are therefore not uniform. We enforce this during the preprocessing of the images to generate a fully masked training and test sets, for both datasets.

Improved random mask However, this process was not optimal to assert the best results possibly, as this complete randomness led to sometimes irrelevant masks covering parts of the image that did not contain interesting objects. That caused the following architectures to perform poorly. Intuitively we expect most of the relevant content of the image to be concentrated in the two-thirds centered frame subset of the original image. A uniform mask is therefore not the best option, as we want to mask actual to cover informative parts of the image. We decided to actually pick the upper left corner of the mask using a Normal distribution of parameters ($\mu = 25$, $\sigma = 10$). Height and width remained drawn uniformly between 10 and 20 pixels. This led to a much improved distribution of the masks over the images.

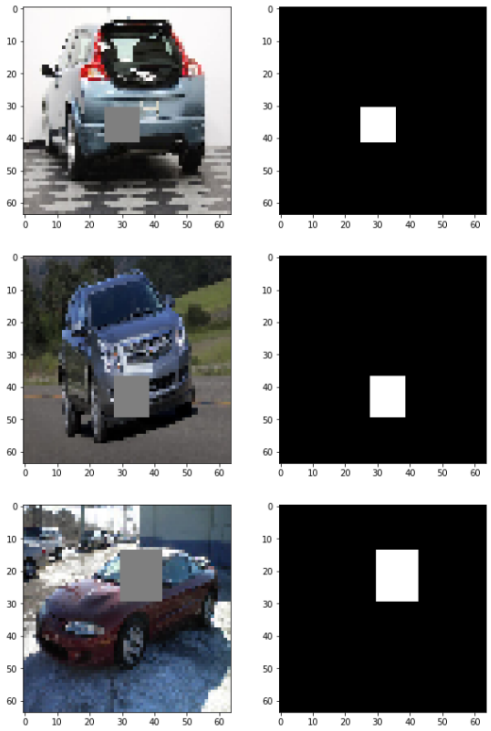

Figure 2: Gaussian positioned masks - Cars dataset

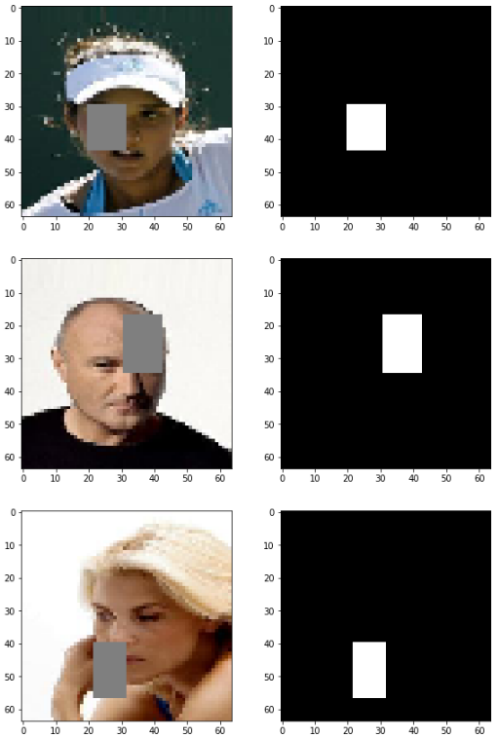

Figure 3: Gaussian positioned masks - Celebrities dataset

3 Attention-based GAN Model

3.1 ApproxNet

Our initial masks were causing the later architecture to perform poorly. This is because a gray-color mask creates a high contrast with the surrounding pixels of the images and fools the attention layer.

We chose to add a mask smoothing net which will allow us a have a more adapted and realistic mask. This neural network takes the masked original image and outputs a deliberately approximate image. We call this feed-forward neural net ApproxNet.

We then apply the equivalent masked-zone of the approximate image as the new mask, which we will feed our GAN with. Thus, we have lowered the contrast between the mask and the image, and such a mask is more realistic for real world applications. We also allow the attention layer to compare the approximate masked region with the surroundings.

Our ApproxNet architecture consists of several dense layers. An architecture involving convolutional layers was indeed too computationally intensive. However, to avoid a consequent overfitting, we have later added dropout layers with rate 0.3. This network contributes to the total generator loss function with an associated reconstruction loss term. This term is the L1 error on the masked region only, and helps building realistic images with the GAN loss, and also avoiding losing context information.

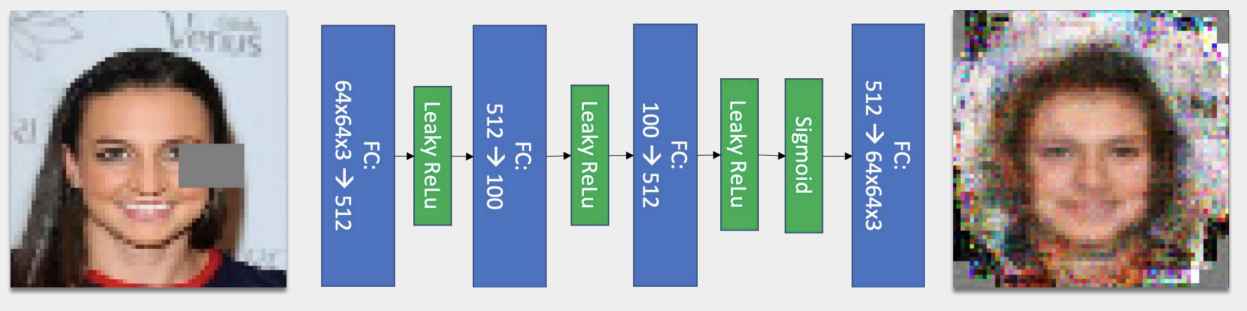

Figure 4: ApproxNet architecture

3.2 Attention-based GAN

Principles of Generative Adversarial Networks

Generative Adversarial Networks (GANs) is a deep generative model of neural nets based on the use of a two-network architecture, with a generator that generates an image and a discriminator to compare real and synthetic images. The discriminator must act as a human trying to classify real from synthetic images. The generator produces samples $x = G(z)$, and the discriminator outputs a probability $y = D(x)$. This is the probability that a sample is a real image: $y = 1$ means that the discriminator is certain that the input image x is a real image and $y = 0$ means that the discriminator is certain that x is a fake sample.

Based on a game theoretic scenario in which the generator network must compete against the discriminator, the formulation of learning is a zero sum game where the payoff of the discriminator is the opposite of the payoff of the generator. Therefore to train the GAN, we use a common loss for both generator and discriminator, and the generator aims at maximizing this loss while the discriminator tries to minimize this loss.

Attention-based GAN advantages

GANs have the ability to generate images from scratch. However, the usual convolutional layers are limited to the information contained in a local neighborhood. Therefore, such an architecture would for example perform poorly at rebuilding the masked wheel of a car.

We can take advantage of another type of layer that helps taking into account the global information in the image: the self-attention layer. Self-attention assigns for each location in the image a set of weights to all the other locations. It tells which locations are most important for a given location. So this time, if we want to rebuild the missing wheel of a car, the network will benefit from the self-attention to get information from another wheel located far away in the image.

The model we used for the Attention layers is inspired by the works of Zhang and al [6] on self-attention generative adversarial networks. It consists of matrix multiplication between different matrices derived from the feature maps (Figure 5).

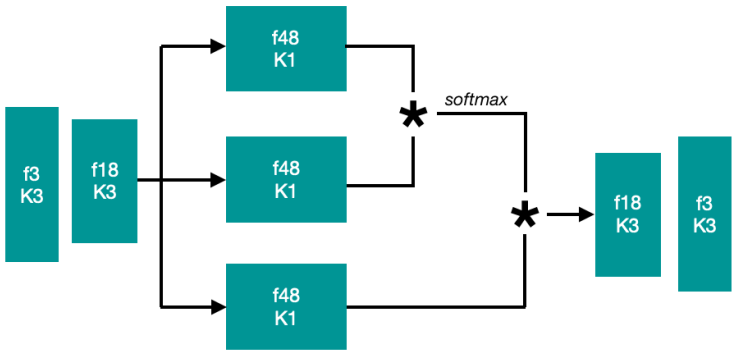

Figure 5: Attention Layers

Discriminator

The discriminator is built with a series of convolutional layers of stride 2, with two final dense layers. It is trained on the the classification cross-entropy loss for real (unmasked) and fake (reconstructed) images:

$$L_D = -\mathbb{E}_{x \in real}[log(D(x))] - \mathbb{E}_{z \in generated}[log(1 - D(G(z)))]$$

3.3 End-to-end Architecture

Here is the overview of our overall architecture :

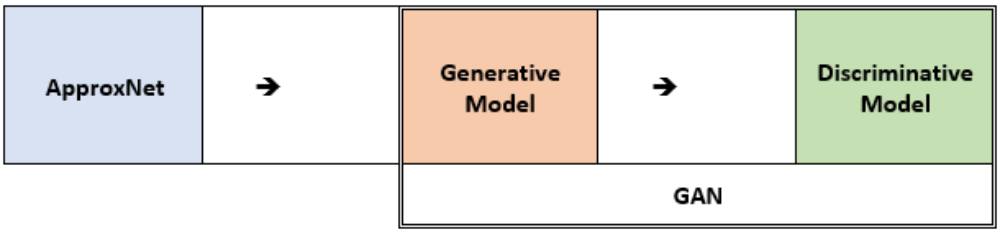

Figure 6: Overall architecture

3.4 Training the GAN

Running the whole model at once can be difficult, as the different layers learn at different paces. Trying to run it as a whole led to layers being very strong and other very weak. We therefore train the model sequentially.

First, and independently from the rest, we train an ApproxNet on 70% of our training data (the remaining 30% being kept as a validation set). Since the whole purpose of this network is to generate an approximation of what lies under the mask, we don't need to minimize the L1 loss perfectly. Instead, we train it for a fixed number of epochs (200 in our experiments).

Then we can jointly train the rest of the generator and the discriminator. To ensure stability during training, we add an L1-loss term to the generator GAN loss, to guide the network to build realistic images. The generator loss therefore becomes:

$$L_G = \mathbb{E}_{z \in generated}[-log(1 - D(G(z))) + \alpha ||G(z) - x||_1]$$

where we tune α on the training set by looking at gradients for each of the L1 and cross-entropy losses.

4 Results

4.1 Baseline Model

We first run a baseline model, computing the SSIM (Structural Similarity) between the original and masked images. This metric translates a similarity in perception by humans, which is more intuitive than a simple pixel wise comparison, using the L2 metric for example.

The average SSIM for the baseline model is 0.9262 on the Celebrities dataset and 0.9268 on the Cars dataset (Figure 7).

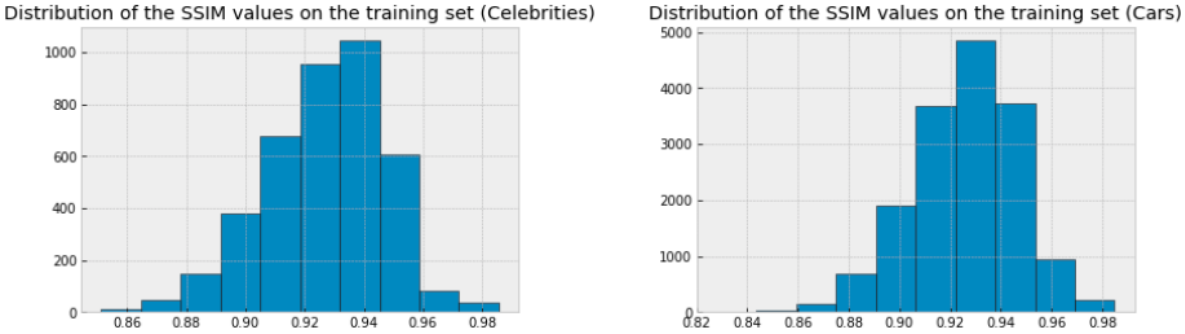

Figure 7: SSIM distribution (baseline model)

4.2 Metrics and Observations

Table 1: SSIM on Cars and Celebrities datasets

Dataset	Baseline	ApproxNet	Full GAN
Cars	0.9262	0.9506	0.9454
Celebrities	0.9268	0.9578	0.9460

On a quantitative aspect, Table 1 summarizes the SSIM scores obtained for the baseline model, for the images generated by ApproxNet and for the full GAN. Both ApproxNet alone and the Full GAN improve on the baseline model. Also, as we will observe on Figure Figure 8, the full GAN produces smoother images. However, because of an offset problem, the Full GAN perfoms slightly less good on the SSIM measure than the AproxNet (Figure 9).

On a qualitative aspect, if the output of ApproxNet looks pixely, we first observe that the generator is pretty good at building coherent and smooth shapes with regard to the rest of the image. The colors are close, but there is still improvement to make since the reconstructed part of the image seems to often be shifted by a constant color (e.g. all pixels are a little red or gray).

4.3 Attention layers

Our experiments show that the attention layer is able to detect and differentiate similar objects of the image. As an example on Figure 10, the image on the right corresponds to the average attention for the pixels in the orange circle on the left image. We observe that the other wheel is highly activated and should help the following layers build a similar wheel.

5 Conclusion and Future Steps

The ApproxNet greatly helped support the attention mechanism. Our experiment without ApproxNet produces images where the mask was still clearly visible (only the pixels near the border with the mask were learned) and somehow fooled the attention with a flat texture recognized in the mask

At the current progress of our experiments, we noticed that the Attention-based GAN is still behind a network directly trained on the L1 Loss. One way to make GAN produce better images is to put more emphasis on the pixels on the border of the mask to make a really smooth transition with the real image pixels.

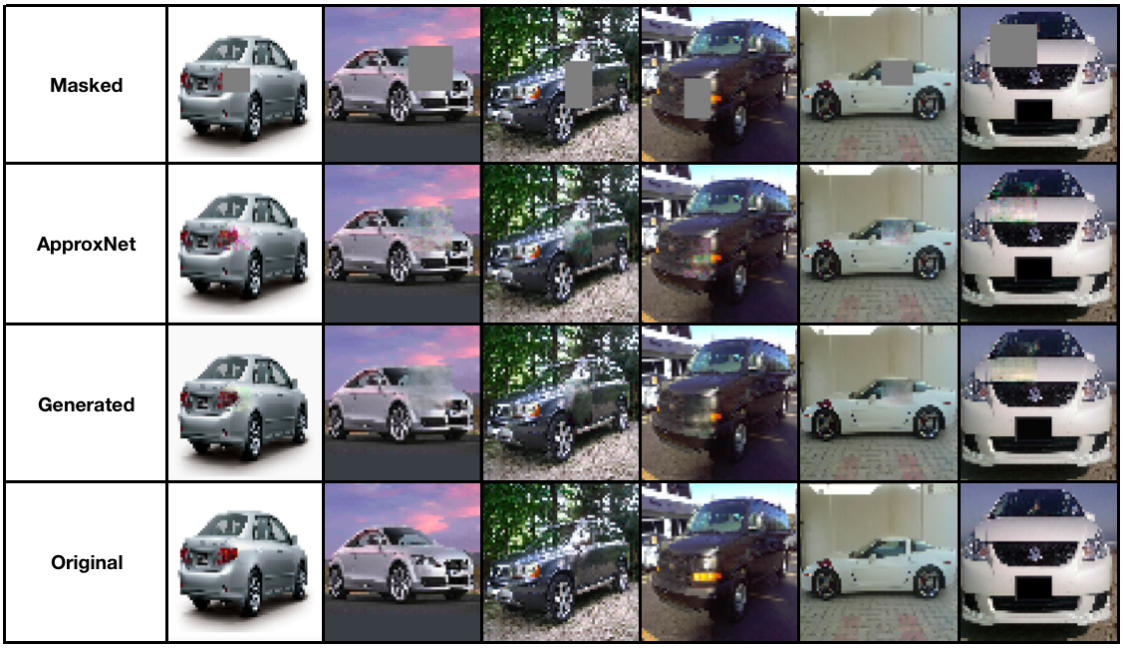

Figure 8: Step-by-step generation of images

Figure 9: Example of reconstruction with offset

Figure 10: Example of self-attention

Future steps would involve scaling up the model, by first using bigger images. In this project, we have limited ourselves to 64*64 pixels images in order to avoid very long computations. Also, with increasing sizes of images, we would have to add more depth to our layers. A final development we would like to try, which can also help the latter point, is to work with pre-trained models for the discriminator or the Encoder-Decoder as to get more accurate results with less iterations without adding too many computations.

6 Lessons learned

Throughout this project, the team has learned three main lessons.

First, the choice of a loss is critical. In our first trials, we trained the GAN using the GAN loss only, which is the cross entropy losses for both generator and discriminator alternatively. This choice led to poor results and the network actually didn't learn how to produce realistic images. This is why we decided to help the generator by adding terms to the loss. These new L1-loss terms on the ApproxNet final layer and the Generator final layer guide the network towards images that resemble the unmasked images.

On a similar note, the training of the GAN is also very sensitive to the initialization and the training program between the generator and discriminator has to be tested empirically. Again, our first trials were tough because the two subnetworks of the generator, the ApproxNet and the Attention Encoder-Decoder, didn't learn at the same pace. We therefore had to manually adjust the number of epochs each net is trained on.

Finally, on a more practical point of view, if we first used Keras to design our models, we quickly realized that Tensorflow would allow us more flexibility to define new layers (like the attention layer). The Tensorflow framework also allows us more control to debug and understand intermediary outputs.

7 Team Contribution

- Remi Denoyer (33%): Contributed to the preprocessing of the data, with a focus on tuning the pipeline and adapting to the Celeb dataset. Involved in training the network, run tests on different losses and architectures.
- Nicolas Hivon (33%): Contributed to the preprocessing of the data, with a focus on the mask and its improvements. Involved in the research reading and design of the GAN and ApproxNet. Participated in deliverables (Poster and both reports).
- Clement Ruin (34%): Contributed to the overall paper research and design of the GAN. Wrote the pipelines for preprocessing images, defined model inputs and add masks to images. Also involved in the training of models and the realization of deliverables.

References

[1] A. Criminisi, P. Perez, and K. Toyama. Region filling and object removal by exemplar-based inpainting. In *Transactions on Image Processing*, volume 13. IEEE, 2004.

[2] Goodfellow I., Pouget-Abadie J., M. Mirza, D. Warde-Farley B. Xu, A. Courville S. Ozair, and Y. Bengio. Generative adversarial networks. *Advances in neural information processing systems*, pages 2672–2680, 2014.

[3] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Generative image inpainting with contextual attention. *arXiv preprint arXiv:1801.07892*, 2018.

[4] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. *4th IEEE Workshop on 3D Representation and Recognition*, 2013.

[5] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2015.

[6] Han Zhang, Ian J. Goodfellow, Dimitris N. Metaxas, and Augustus Odena. Self-attention generative adversarial networks. *CoRR*, abs/1805.08318, 2018.